

# Evaluation of Vector Sensors for Adaptive Equalization in Underwater Acoustic Communication

by

Matthew Robert Lewis

B.S., Rensselaer Polytechnic Institute (2007)

Submitted in partial fulfillment of the requirements for the degree of  
Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

WOODS HOLE OCEANOGRAPHIC INSTITUTION

September 2014

© Matthew Robert Lewis, MMXIV. All rights reserved.

The author hereby grants to MIT and WHOI permission to reproduce and to distribute  
publicly paper and electronic copies of this thesis document in whole or in part in any  
medium now known or hereafter created.

Author.....  
Department of Mechanical Engineering and  
Department of Applied Ocean Science and Engineering  
August 8, 2014

Certified by.....  
Dr. James C. Preisig  
Associate Scientist with Tenure  
Thesis Supervisor  
Woods Hole Oceanographic Institution

Accepted by .....  
Dr. Henrik Schmidt  
Professor of Mechanical and Ocean Engineering  
Chair, Joint Committee for Applied Ocean Science and Engineering  
Massachusetts Institute of Technology

Accepted by .....  
Dr. David E. Hardt  
Professor of Mechanical Engineering  
Chair, Mechanical Engineering Committee on Graduate Students  
Massachusetts Institute of Technology



# Evaluation of Vector Sensors for Adaptive Equalization in Underwater Acoustic Communication

by

Matthew Robert Lewis

Submitted to the Department of Mechanical Engineering and  
Department of Applied Ocean Science and Engineering  
on August 8, 2014, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Oceanographic Engineering

## Abstract

Underwater acoustic communication is an extremely complex field that faces many challenges due to the time-varying nature of the ocean environment. Vector sensors are a proven technology that when utilizing their directional sensing capabilities allows us to minimize the effect of interfering noise sources. A traditional pressure sensor array has been the standard for years but suffers at degraded signal to noise ratios (SNR) and requires maneuvers or a lengthy array aperture to direction find. This thesis explores the effect of utilizing a vector sensor array to steer to the direction of signal arrival and the effect it has on equalization of the signal at degraded SNRs. It was demonstrated that utilizing a single vector sensor we were able to steer to the direction of arrival and improve the ability of an equalizer to determine the transmitted signal. This improvement was most prominent when the SNR was degraded to levels of 0 and 10 dB where the performance of the vector sensor outperformed that of the pressure sensor in nearly 100% of cases. Finally, this performance improvement occurred with a savings in computational expense.

Thesis Supervisor: Dr. James C. Preisig  
Title: Associate Scientist with Tenure  
Woods Hole Oceanographic Institution





# Dedication

*For my wife* MEGHAN

*Your unwavering love, support and encouragement has been my rock as school work  
continually trumped everything else!*

*And* OUR UNBORN CHILD.

*Your mother and I are anxiously awaiting your arrival!*



## Acknowledgments

First and foremost, I would like to thank my advisor Dr. Jim Preisig. Without his unending patience, guidance, assistance and support this thesis would never have been completed. And, for believing in me and jumping through hoops to ensure that I got the best education possible. Finally, for his commitment to quality research that ensured that this product was the best it could be. It has truly been a pleasure to work with you.

Thanks to Jen, my partner-in-crime, for her friendship and help in navigating all the travails that MIT and WHOI had to throw our way.

To Jon and Sheida, thank you for your friendship and infinite patience in the day-to-day struggle of life as a graduate student.

To Sam, thank you for allowing me the chance to pursue this opportunity when you alone could have prevented it. And for always seeing in me that which I may not have seen in myself.

I would also like to thank the United States Navy, the Oceanographer of the Navy and the Naval Postgraduate School for the opportunity to participate in this program.

And finally, I would like to thank my incredible family for your support and love through it all.

This thesis would not have been possible without support from the Office of Naval Research (through ONR Grants #N00014-11-10426 and #N00014-07-10738).



# Contents

<b>1</b>	<b>Introduction</b>	<b>19</b>
1.1	Motivation . . . . .	20
1.1.1	Challenges in Underwater Acoustic Communications . . . . .	21
1.1.2	Need for Adaptive Equalization . . . . .	22
1.2	Underwater Communications Systems . . . . .	22
1.3	Vector Sensors . . . . .	24
1.4	Thesis Organization . . . . .	25
<b>2</b>	<b>Background</b>	<b>27</b>
2.1	Notation . . . . .	27
2.2	Vector Sensors . . . . .	27
2.2.1	Theory . . . . .	28
2.2.2	Advantages . . . . .	33
2.2.3	Types of Vector Sensors . . . . .	40
2.3	Adaptive Equalization . . . . .	42
2.3.1	System Model . . . . .	43
2.3.2	Method of Least Squares . . . . .	46
2.3.3	Recursive Least Squares Algorithm . . . . .	50

2.3.4	Time-Update Recursive Least Squares Algorithm . . . . .	52
2.3.5	Multiple Channels . . . . .	57
<b>3</b>	<b>Modeling</b>	<b>59</b>
3.1	BELLHOP and Ray-based Models . . . . .	59
3.2	Inputs . . . . .	64
3.3	Results and Interpretation . . . . .	69
<b>4</b>	<b>The Experiment</b>	<b>77</b>
4.1	Introduction . . . . .	77
4.2	Physical Geometry . . . . .	78
4.3	Time Variability . . . . .	78
4.4	Data Structure . . . . .	79
4.5	Noise . . . . .	83
<b>5</b>	<b>Algorithms and Processing</b>	<b>89</b>
5.1	Introduction . . . . .	89
5.2	Received Signal Processing . . . . .	90
5.3	Method . . . . .	96
5.4	Equalization Parameters . . . . .	102
5.4.1	Filter Length . . . . .	102
5.4.2	Forgetting Factor and Step Size . . . . .	103
5.5	Results . . . . .	106
5.5.1	Signal-to-Noise Ratio . . . . .	106
5.5.2	Adaptive Equalization . . . . .	110
5.5.3	Computational Cost . . . . .	113

<b>6</b>	<b>Conclusions</b>	<b>117</b>
6.1	Summary and Conclusions . . . . .	117
6.2	Future Work . . . . .	120





# List of Figures

1-1	Block diagram of communications system . . . . .	23
2-1	Geometry of vector sensor . . . . .	31
2-2	Beampattern of 10-element conventional pressure sensor array steered to $\phi = \frac{\pi}{2}$ . . . . .	35
2-3	Vector sensor modulation term, $B_v$ . . . . .	36
2-4	Beampatterns of 10-element conventional array ( $B_p$ ), vector-sensor modulation term ( $B_v$ ) and the combined beampattern ( $B_{comb}$ ) steered to $\frac{\pi}{2}$ . This is a modified version of the plot shown in [13] . . . . .	38
2-5	Beampatterns of 10-element conventional array ( $B_p$ ), vector-sensor modulation term ( $B_v$ ) and the combined beampattern ( $B_{comb}$ ) being steered to $\frac{\pi}{3}$ (a), $\frac{\pi}{4}$ (b) and $\frac{\pi}{6}$ (c). This is an expansion of the plots shown in [13] . . . . .	39
3-1	Reflection and transmission at a fluid-fluid interface . . . . .	60
3-2	Ray refracting through a stack of fluid-fluid interfaces . . . . .	62
3-3	Sound speed profile of KAM11 from JD 176-182 . . . . .	65
3-4	Sound speed profile used for BELLHOP simulations . . . . .	66

3-5	(a) Estimated sound speed profile (b) Ray trace of estimated sound speed profile . . . . .	68
3-6	Ray traces for (a) source at 40 meters and (b) source at 50 meters . .	70
3-7	Eigenray traces for (a) source at 40 meters and (b) source at 50 meters	72
3-8	Arrivals with respect to time at the array for a 50 m source . . . . .	73
3-9	Tranmission loss plots for (a) source at 40 meters and (b) source at 50 meters . . . . .	75
4-1	Normalized channel impulse response of the (a) pressure, (b) x-velocity, (c) z-velocity and (d) y-velocity elements of a single sensor of the vector sensor array. . . . .	80
4-2	Spectrogram of one repetition of the transmitted signal . . . . .	81
4-3	Spectrograms from segment 2 which was used as additive noise: (a) Pressure, (b) X-velocity, (c) Y-velocity and (d) Z-velocity . . . . .	84
4-4	Energy of noise file in the frequency band of the signal . . . . .	85
5-1	Spectrograms from segment 86: (a) Pressure, (b) X-velocity, (c) Z-velocity and (d) Y-velocity . . . . .	91
5-2	Spectrograms from segment 101: (a) Pressure, (b) X-velocity, (c) Z-velocity and (d) Y-velocity . . . . .	92
5-3	Spectrograms from segment 104: (a) Pressure, (b) X-velocity, (c) Z-velocity and (d) Y-velocity . . . . .	93
5-4	Energy in the band of 2.5-5.5 kHz from segment 104 of received data	94
5-5	Matched filter output of base banded, filtered and down sampled received pressure signal of (a) segment 86 - sensor 1, (b) segment 86 - sensor 4, (c) segment 101 - sensor 1, (d) segment 101 - sensor 4, (e) segment 104 - sensor 1 and (f) segment 104 - sensor 4 . . . . .	97

5-6	Array resolution of sensor 4 from segment 104 (a) shows the array resolution of the velocity sensor in the horizontal plane when resolving the ambiguity while (b) does not resolve the ambiguity . . . . .	99
5-7	Array resolution of segment 101, sensor 1 in the horizontal plane of the velocity sensor with the chosen optimum angle overlaid in white .	100
5-8	Optimum angles for sensor 1 of segment 86. The upper plot shows the optimum horizontal angle ( $\phi$ ) and the bottom plot shows the optimum elevation angle ( $\theta$ ) . . . . .	101
5-9	Equalizer BER performance for different averaging window lengths . .	103
5-10	Examples of interpolated results to determining optimum $\lambda$ and $\mu$ values. The optimum in each run is shown by the white star and explicitly stated in table 5.2 . . . . .	105
5-11	Block diagram of our processing methodology . . . . .	107
5-12	SNR of velocity components of each segment analyzed for this thesis (a) shows the segments received on sensor 1 and (b) shows the segments received on sensor 4 . . . . .	108
5-13	Comparison of the SNR of beamformed velocity with multi-channel velocity and its components . . . . .	109
5-14	CDF of BER for native SNR . . . . .	111
5-15	CDF of BER for SNR = 20 dB . . . . .	111
5-16	CDF of BER for SNR = 10 dB . . . . .	112
5-17	CDF of BER for SNR = 0 dB . . . . .	112



# List of Tables

2.1	Notations used in the thesis . . . . .	28
2.2	Adapted Version of Haykin's RLS Algorithm . . . . .	52
2.3	Adapted Version of TU-RLS Algorithm . . . . .	56
4.1	Structure of the transmitted signals . . . . .	82
5.1	Average signal-to-noise ratio of data segments expressed in dB . . . . .	95
5.2	Optimum $\mu$ and $\lambda$ values for data combinations analyzed . . . . .	105



# Chapter 1

## Introduction

*We are tied to the ocean. And when we go back to the sea - whether it is to sail or to watch it - we are going back from whence we came.*

- John F Kennedy

The human race has always looked to push the bounds on our ability to understand the world around us. There are few environments in the world that are more inhospitable to human life and our technology than the ocean. Yet with each passing day we strive to improve our ability to operate over the whole range of ocean environments. Whether it be with submarines, unmanned autonomous vehicles or habitats on the ocean floor where human beings live for days at a time we are continually testing and striving to improve our capabilities.

Our capacity to communicate over distances in the ocean is one of the restrictive limits on our ability to advance undersea technology. Every advance to improve the range over which we can communicate wirelessly as well as the robustness of that method of communication opens up a new front of exploration for many fields in the ocean science community.

This thesis explores a proven piece of technology: vector sensors. In doing so we hope to show that their use in underwater acoustic communications can improve our ability to successfully communicate in a variety of environments. An example of an environment that can be improved through the use of vector sensors is a low signal-to-noise ratio (SNR) environment whether it be due to high background noise or low signal power.

In this chapter we will introduce the motivation for the problem we are attempting to solve. We will also present some of the challenges facing underwater acoustic communications as well an overview of how underwater communications systems function. This chapter will conclude with an overview of how this thesis is organized.

## 1.1 Motivation

The goal of underwater communications is just like any other type of wireless communication; transmit a data packet over a distance ranging from a few hundred meters to a few hundred kilometers. Most wireless communications at the earth's surface utilize radio frequency (RF) communications but electromagnetic (EM) waves at very low frequencies which propagate further in seawater than higher frequency EM waves still only have a range in the ocean on the order of only ten meters. This makes using electromagnetic waves in underwater communications unfeasible. Optical communications have similar range issues under the surface of the ocean except they are also significantly hindered by murky water.



### 1.1.1 Challenges in Underwater Acoustic Communications

Acoustic waves are the accepted medium for underwater communications as they provide reasonable signal levels at ranges of interest. However, underwater acoustic communications are widely viewed as the most perilous and difficult communications tool in use today. There are three major challenges when working with underwater communications: (1) attenuation and noise, (2) multiple path and (3) Doppler. [26] The combination of all of these result in complicated channel impulse responses with time-varying inter-symbol interference (ISI) and Doppler effects. [19]

Due to this variability there is no particular model that can be used to determine the effect that each of these will have on a particular experiment. The acoustic channel must be analyzed and parameters optimized for each individual data set that is analyzed. In most cases an improperly modeled channel will significantly degrade the results of further analysis [19].

A subset of signal processing called model-based processing makes the assumption that you know the make-up of the channel and are able to model it correctly. Considering the case where you are able to properly model the channel you can then utilize optimal detectors. A matched filter, or the time-reversed signal, results in the maximal output SNR with a coherent processing gain of the time-bandwidth product of the signal [20]. Another optimal detector would be a linear minimum-mean squared error (MMSE) estimator which utilizes the orthogonality principle to determine the optimal filter coefficients which are used to estimate the channel. In order to use these optimal detectors you must have a priori knowledge about the channel, which is often not the case. Without a priori statistics describing the channel you are forced to adjust the filter to fine tune those statistics on the fly, this is called adaptive equalization.

### 1.1.2 Need for Adaptive Equalization

The time varying ISI mentioned in the previous section is one of the main motivations for the use of adaptive equalizers at the receiver to establish reliable communications links. An equalizer is a set of linear filters that compensate for intersymbol interference and ambient noise in the received signal to determine the signals that were transmitted. The result of the adaptation process that the equalizer undergoes is a set of filter coefficients that similarly must be varied due to the time-variability of the channel. The output of the equalizer is a decision, either soft or hard that estimates the transmitted symbol. This is done in many applications besides underwater communications, for example in mobile radio communications [21, 22] and magnetic disk storage channels [2].

## 1.2 Underwater Communications Systems

It is important to have a basic understanding of the development of underwater acoustic communications and their uses in current systems. Underwater communications were likely developed as a necessity during World War II as a method of communicating with and between submarines. The development of digital signal processors along with many other advancements has significantly improved both the range and data throughput in underwater acoustic communication systems [25]. This overview of underwater communication systems will be brief in nature an extensive review can be found in [25] as well as descriptions of advances and the state of the art in [24, 12].

Since the early 1980's acoustic communications capabilities have improved more than eighty-fold and two-hundred fold in shallow and deep water, respectively. These

advances have come in many different areas of research whether through coherent or incoherent modulation, improved processing capabilities supporting more complex receiver algorithms or improved error control [12]. All of these advances are limited by our ability to properly model the sound channel specifically with regards to time variability in the channel. Currently the computational expense to model a varying sound channel is extremely high and even if that expense were worth the cost there would not be the requisite amount of data necessary to create a robust model. As a result we must incorporate new technologies and continually more efficient processing algorithms that help us to account for the time-varying nature of the ocean environment.

While there are many different systems as well as communication and signal processing techniques utilized we will be focusing on a system described by the basic block diagram seen in figure 1-1. More specifically our focus will be directed to the

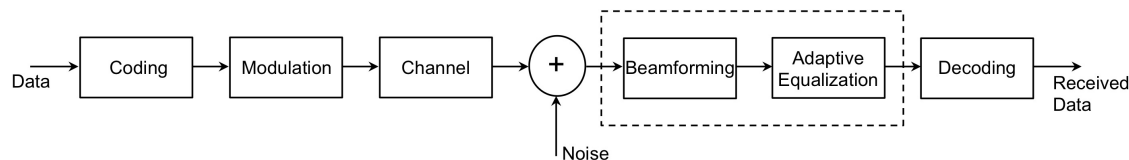


Figure 1-1: Block diagram of communications system

structures on the receiving end of the signal. The goal of each of these blocks is to minimize the impact of the not only ISI but noise on the proper detection of the signal.

All of the inputs and functions of blocks are controlled except for two: the noise and the channel. The first two blocks are the creation of the signal from the data and the modulation of that signal so that it can be transmitted. The signal is then transmitted through the unknown channel with an unknown amount of additive noise combined with the signal. The beamforming block can take on different purposes

depending on the specific type of array being utilized (i.e. line-array of hydrophones or a single vector sensor). The adaptive equalization block is the block that attempts to extract the transmitted signal from the channel output by estimating the channel. Finally the decoding block takes the equalizer output and turns it into our estimation of the received data.

### 1.3 Vector Sensors

Traditional acoustic communication systems have relied upon a single hydrophone, a string of hydrophones or a spatial array of hydrophones to detect the transmission. The hydrophone we are describing is an omni-directional sensor that detects the local pressure fluctuations caused by the propagating acoustic signal - it is in essence an underwater microphone. Omni-directional hydrophones are extremely common due to their ease of manufacture, maintenance, use and analysis. Decades of experience both in the scientific and military community have also shown that they stand up extremely well in the destructive ocean environment.

Commonly a string of omni-directional hydrophones are formed into an array that can be mounted to the side of a ship, towed behind a body moving through the ocean or deployed on the seabed. In most applications these arrays are uniformly spaced line arrays due to the ease of deployment, processing and subsequent analysis. However, it is the omni-directional nature of the hydrophones mounted on these arrays that prevents them from immediately determining the direction of the incident pressure field. The processing of these one-dimensional horizontal or vertical line arrays can only provide an azimuth or elevation to the source with a left/right or up/down ambiguity around the axis of the array that requires further analysis to resolve.

When analyzing any configuration of sensors, clearly the more information you

are able to detect the better the performance of the sensor will be. This is the same case when we look at acoustic sensing, the pressure field given by a omni-direction hydrophone is a lot of data but if we are also able to obtain the direction of the sound waves that is better. This is what a vector sensor does, it takes one omni-directional hydrophone to measure the pressure field and three orthogonal sensors to measure the component velocities of the incident sound waves. The component velocities allow allow a single vector sensor to estimate the direction of propagation of a received signal.

## 1.4 Thesis Organization

This thesis is organized in the following manner: in Chapter 2, we will provide a detailed introduction to not only vector sensors but also adaptive equalization and the RLS algorithm.

Chapter 3 will describe Bellhop modeling that was performed to get a better idea of the nature of transmission path of the KAM11 experiment. It will describe the assumptions that were made and the effects that it has on further data analysis.

In Chapter 4 we will describe the data set that was used for our analysis. Specifically the equipment that was used as well as the transmitted communication sequence. We will also analyze the noise that was present in the given environment and to describe the additional noise that was added to our received signals to degrade the SNR.

Chapter 5 will analyze the algorithm that we developed to jointly process pressure and velocity data from a vector sensor. It will step through the process of determining the optimal angle to beamform the velocity components. It will then show the results of both a single channel equalizer of beamformed velocity data compared to just

pressure data or velocity data that was not beamformed.

Finally, Chapter 6 will summarize the results presented in this thesis as well as to provide some insights for areas of development for future work.

# Chapter 2

## Background

### 2.1 Notation

Table 2.1 contains notation that is used throughout this thesis and will not be redefined.

### 2.2 Vector Sensors

As mentioned in the previous chapter vector sensor technology has been around for some time [14]. However, until recently their design was such that they could not be using in existing military systems due to the size of the accelerometers. With the development of piezoelectric crystal accelerometers vector sensors can now fit into existing military towed array systems and meet the performance requirements necessary for their use [23]. Due to this the amount of research on them has picked up significantly. This section will focus on the basic theory of how a vector sensor works, a more detailed discussion on the advantages of a vector sensors and the

Symbol Type	Meaning
$x$ (non-boldface math symbol)	Scaler constant or variable
$\mathbf{u}$ (boldface, lowercase symbol)	Column vector
$\mathbf{A}$ (boldface, uppercase symbol)	Matrix
$\mathbf{I}_N$	$N \times N$ identity matrix
$\circ$	Hadamard product of two vectors
$*$	Complex conjugate
$T$	Matrix transpose
$H$	Matrix Hermitian

Table 2.1: Notations used in the thesis

different kinds of vector sensors that are available.

## 2.2.1 Theory

### Assumptions

As with all scientific research there are a number of assumptions that are made to simplify the processing of vector sensors. This section describes the common assumptions and the reasoning behind them [15, 13]. When referring to the maximum and minimum wavelength these are determined by the minimum and maximum frequency of the transmitted signal spectrum, respectively.

(1) *Plane wave propagation.* This implies that all the requirements of the far-field assumption are met which include that the incident sound waves are planar at the time they reach the sensors; this implies that the unit vector pointing towards the source is approximately independent of the sensor position on the array. This requires that the distance from the source to the array is much greater than the both the length of the array and the maximum wavelength of the transmitted signal. Additionally it assumes that the source is a point source where the source size is much



smaller than the distance between the source and the sensor. Finally, it requires that the environment is homogeneous in the vicinity of the array to ensure there is no refraction across the array. The plane wave equation can be seen below in equation (2.1) where it relates pressure  $p$  across time  $t$  and position  $\mathbf{r}$  to the sound speed  $c$  while  $\mathbf{u}$  is unit vector pointing towards to source. The function  $f$  is the complex envelope of the signal propagating through the environment.

$$p(\mathbf{r}, t) = f\left(\frac{\mathbf{u}^T \mathbf{r}}{c} + t\right) \quad (2.1)$$

(2) *Band-limited and narrowband signals.* The band-limited assumption requires that the transmitted signal is band-limited such that it has a spectrum including only frequencies  $\omega$  such that  $\omega_{min} \leq |\omega| \leq \omega_{max}$  where  $0 < \omega_{min} < \omega_{max} < \infty$ .

The narrowband assumption is concerned with the delay in arrivals at each sensor of an array. Arrivals at the  $i^{th}$  and  $j^{th}$  sensors are defined as  $P_i(t)$  and  $P_j(t)$  where one is merely a phase-shifted versions of the other.  $P(t)$  is the complex envelope of the acoustic pressure at time  $t$  and the maximum delay in arrival time across the length of the array is defined as  $\tau_{max}$ . This assumption requires that the following relationship is satisfied  $P(t) \simeq P(t - \tau_{max})$ . Finally, the maximum bandwidth which is defined as  $\omega_{bw} = \omega_{max} - \omega_{min}$  must be much less than the inverse of the maximum delay as shown in equation (2.2).

$$\omega_{bw} \ll \frac{1}{\tau_{max}} \quad (2.2)$$

The narrowband assumption is consistent with the model in [13] however is not required for this thesis because all processing performed was single sensor processing.

(3) *Co-located sensor components.* This indicates that the three-axes accelerom-

eters and the omni-directional hydrophone are located at the same point. For this to be satisfied it requires that the distance between each of the sensor components be much less than the minimum wavelength.

(4) *Point sensors.* This assumption is very similar to the co-located sensor components with a slight variation. It takes into account that the sensor can be modeled as a single point. This means that the sensor dimensions is much smaller than the minimum wavelength of the incident signal wave.

(5) *Orthogonal accelerometers.* This assumes that the axes of the accelerometers of the vector sensor are all orthogonal to each other. This is standard in the production of vector sensors.

## Geometry

The geometry of a vector sensor clearly will depend on the sensors that are included in that particular vector sensor. In this thesis we will be focusing on a construction that include three orthogonally oriented accelerometers plus a omni-directional hydrophone co-located with the accelerometers. The construction of our model is based upon that described in [13] where a more detailed explanation can be found. It should be noted that we have modified the framework proposed by Kitchens to match our terminology and variables. This results in a geometry that is displayed in figure 2-1 which has a 4 x 1 array manifold vector as seen in equation (2.3) [15].

$$\mathbf{a}_k = \begin{bmatrix} \sin \theta_k \cos \phi_k \\ \sin \theta_k \sin \phi_k \\ \cos \theta_k \\ \eta \end{bmatrix} = \begin{bmatrix} u(\theta_k, \phi_k) \\ v(\theta_k, \phi_k) \\ w(\theta_k) \\ \eta \end{bmatrix} \quad (2.3)$$

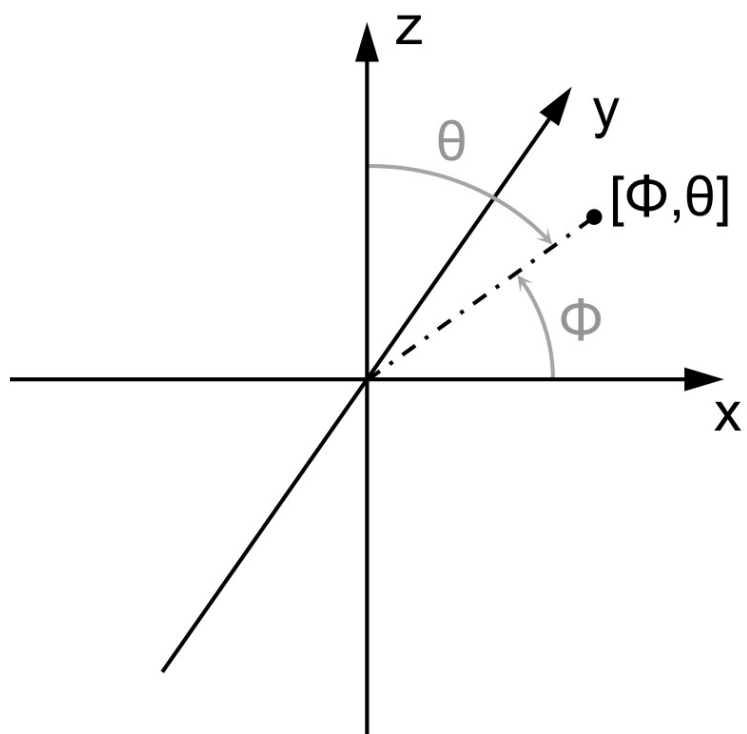


Figure 2-1: Geometry of vector sensor

where  $0 \leq \theta_k \leq \pi$  denotes the  $k^{\text{th}}$  source's arrival elevation measured from the positive vertical axis and  $0 \leq \phi_k \leq 2\pi$  symbolizes the  $k^{\text{th}}$  source's azimuth angle measured counter-clockwise from the positive  $x$  axis. In this thesis we are only dealing with one source but this notation is included for extension to multiple source environments. The first three components in equation (2.3) represent each of the  $x$ ,  $y$  and  $z$  components of the velocities determined using each of the accelerometers and thus give the Cartesian direction of the incident direction of arrival.

The final component in  $\mathbf{a}_k$  represents the pressure signal and is called the polarization term. This term is a normalization factor that includes the gain difference between the pressure sensor and the velocity sensors. As this is a factor relating pressure to velocity it includes a factor of  $\frac{1}{\rho c}$  which will be shown in equation (2.8) to relate pressure and velocity [13]. The normalization term would thus be  $\frac{1}{\rho c}$  if the gain was equal between pressure and velocity sensors. The normalization term would then tend towards 0 if the gain in the velocity sensors is much greater than the pressure sensor. Clearly in the converse case where the gain in the pressure sensor is much greater than the velocity sensors the normalization term will increase and become greater than  $\frac{1}{\rho c}$ . The effects of these changes in the normalization terms will be shown and discussed in the next section.

As described in [28] the presence of the pressure hydrophone is important in the setup due to the ambiguity that exists in which hemispherical side of the array that the signal arrives in (i.e. the top hemisphere or bottom). Additionally the pressure hydrophone resolves the  $180^\circ$  ambiguity which is characteristic of accelerometers and other velocity sensors. The pressure sensor accomplishes this by allowing us to distinguish between acoustic compressions and dilations which is able to inform us of the proper direction of arrival, this is discussed further in section 5.3.

### 2.2.2 Advantages

Vector sensors are desirable for use due to one major characteristic: directional sensing. Delving further into that statement to understand why directional sensors are appealing breaks down into two capabilities: (1) ability to selectively reject the noise emitted from discrete interfering noise sources and (2) the ability to localize targets [23]. On the surface this may not seem like a significant gain but in reality it makes all the difference in the world. When using an omni-directional hydrophone alone a loud interfering signal coming from a source in a direction different from the source of a desired signal can significantly degrade the ability to effectively process the desired signal.

Consider using a single vector sensor to determine the direction from which a signal is propagating and then steering the vector sensor to that direction. When steering to the direction of the source of a desired signal, the processor can reduce the effect of interfering signals coming from other directions. This is done by linearly combining the signals from the three velocity sensor based upon trigonometric relationships. The weights are chosen to pass signals from the desired arrival direction without attenuation while attenuating signals from other directions.

Another way to look at this advantage is to compare an array of vector sensors with a traditional pressure-sensor array. Array processing of a traditional pressure-sensor array is dependent on a variety of techniques that depend on propagation delays between each sensor. While the traditional array requires a physical aperture to resolve or discriminate between signals with different propagation directions, the vector sensor requires only the small aperture of a single vector sensor to do this [5]. The directional information from one vector sensor has been shown to identify up to two sources in [7] that would not be possible with a single pressure sensor but would

require a line array of multiple sensors.

The major advantage that was mentioned in section 1.3 is the ability to resolve ambiguity in uniform line array (ULA). Here it is important to step back and understand some of the limitations of an ULA of omni-directional hydrophones. A linear array of hydrophones due to the symmetry of the array around its axis result in a response in the form of conical angles. This results in ambiguity in the direction of arrival with respect to the left or right side of the array (port/starboard). For a uniform line array oriented along the x-axis the pressure sensor beampattern is given by equation (2.4)

$$B_p = \frac{1}{N} \frac{\sin\left(\frac{N}{2} \frac{2\pi d}{\lambda} (\cos \phi - \cos \hat{\phi})\right)}{\sin\left(\frac{1}{2} \frac{2\pi d}{\lambda} (\cos \phi - \cos \hat{\phi})\right)} \quad (2.4)$$

where  $B_p$  is the magnitude of the array response,  $N$  is the number of elements in the array,  $d$  is the inter-element spacing which for convenience is normally defined in terms of the wavelength  $\lambda$ . Finally  $\phi$  is the azimuth angle of the array and  $\hat{\phi}$  is the angle the array is being steered, both of the angles are relative to the direction the array is pointing. If you are sensing a signal coming from an angle of  $\frac{\pi}{2}$  that same response would also be seen coming from an angle of  $\frac{3\pi}{2}$  or  $\frac{-\pi}{2}$ , this is best demonstrated in figure 2-2 which shows the beampattern of a 10 element omni-directional pressure sensor array being steered to an angle of  $\frac{\pi}{2}$ . In this thesis our convention will be that an azimuth angle of  $\phi = 0$  is forward endfire of the array and an azimuth angle of  $\phi = \frac{\pi}{2}$  is port broadside. In figure 2-2 you can then see that the beampattern is symmetric across the axis of the array

The only way to resolve the ambiguity of whether or not the source is on the port or starboard side of the array is to either get information from another source or to maneuver the array to another location so that you can determine the “true”

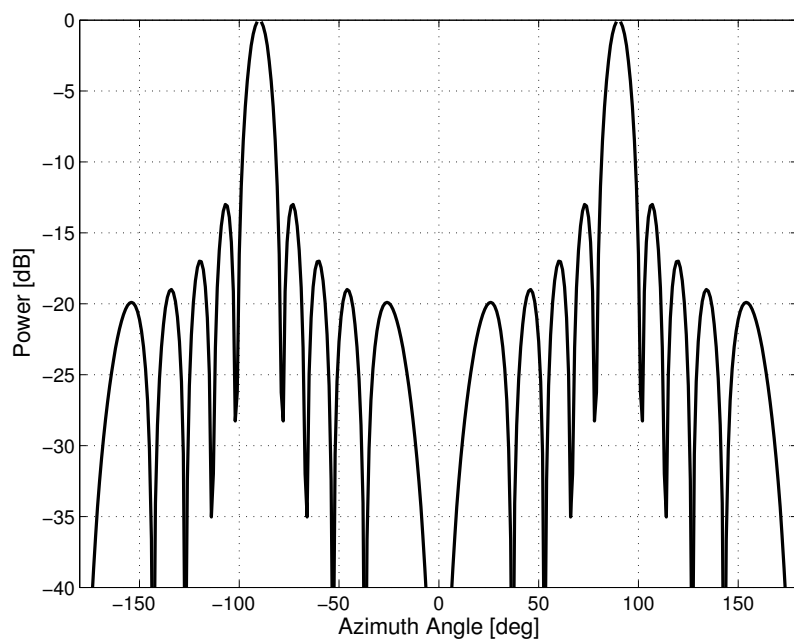


Figure 2-2: Beampattern of 10-element conventional pressure sensor array steered to  $\phi = \frac{\pi}{2}$

location of the source where the conical angles cross. This is a significant theoretical and operational limitation of a linear array of omni-directional hydrophones. In a line array of vector sensor the velocity components are not symmetrical around any axis of the array and therefore are capable of “resolving” the ambiguity that is present in a traditional pressure-sensor array [13].

The relationship shown in equation (2.5) was derived in [13] and is called the vector sensor modulation term where  $\eta$  is a normalization constant and  $\psi$  in three dimensions is the angle between the azimuth angle of a source and the steering angle.

$$B_v \triangleq \left| \frac{\eta^2 + \cos \psi}{\eta^2 + 1} \right|^2 \quad (2.5)$$

A plot of the modulation term is shown in figure 2-3. As the azimuth angle moves

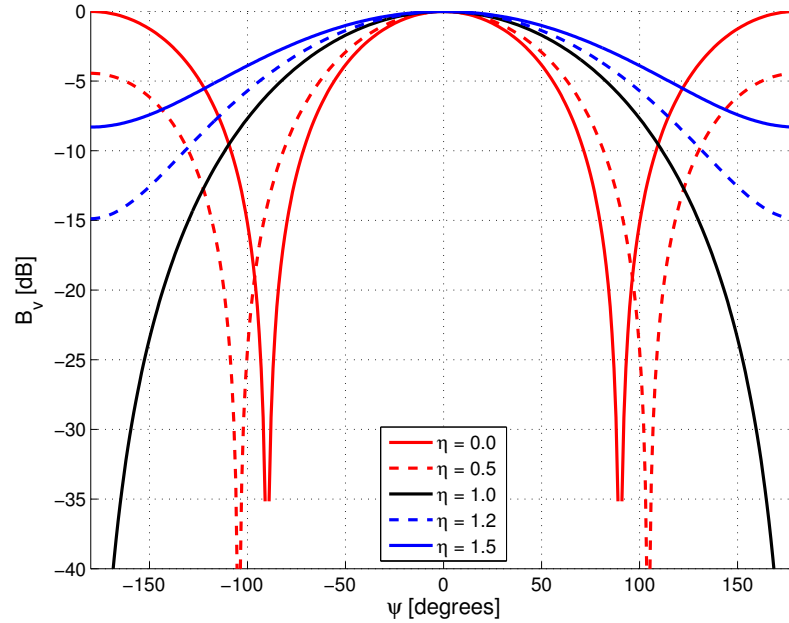


Figure 2-3: Vector sensor modulation term,  $B_v$

away from the steering angle this will cause  $\psi$  to increase and the cosine to get smaller



and therefore the modulation term will decrease as well. When this modulation term is then applied to the traditional pressure sensor beampattern it will apply a lesser weighting to the ambiguous arrival, but leave the true arrival at the same level. It is important to take note of the effect the normalization term has, with the proper normalization of data,  $\eta = 1.0$ , it gives an ideal null at  $\psi = \pm\pi$ . If the data is not properly normalized as the gain decreases ( $\eta \rightarrow \infty$ ) the modulation terms moves towards unity and starts to mirror the performance of a pressure-sensor array [13]. We now expand upon what was covered in [13] and consider the case where the gain of the velocity sensors is greater than the pressure sensors (i.e.  $\eta < 1.0$ ). You can see in figure 2-3 that as the normalization term approaches zero; or the gain of the velocity sensors becomes larger than the gain of the pressure sensors; the beampattern of the modulation term shows the  $180^\circ$  ambiguity that comes from the velocity-only data of the vector sensor. This tells us that if the gain in one element dominates the other than the whole sensor will start to mimic the element whose gain is much greater.

The conventional beam and vector sensor modulation terms are combined in point-wise multiplication as represented by equation (2.6).

$$B_{comb}(\psi) = B_p(\psi) B_v(\psi) \quad (2.6)$$

Due to the width of the modulation term envelope; which is seen overlaid with the pressure sensor beampattern in figure 2-4; in the region where there is no ambiguity (i.e.  $\cos \psi \approx 1$ ) the response will be dominated by the pressure-sensor response. The vector sensor term has its effect on the side-lobe and ambiguity regions. This can also be seen in figure 2-4 which shows the combined beampattern with a null in the ambiguity region and depressed side-lobes around the main lobe.

The case in figure 2-4 is one where the array is steered to  $\frac{\pi}{2}$  and the ambiguous

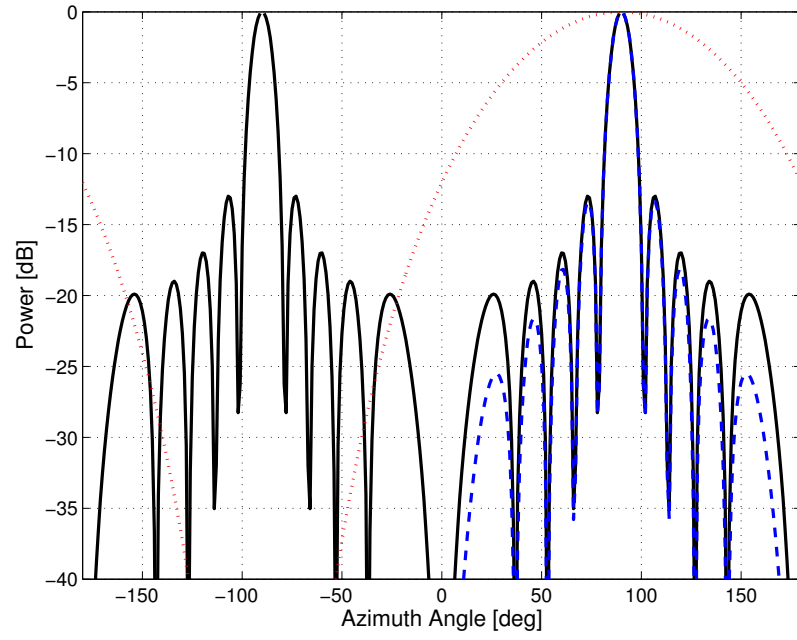


Figure 2-4: Beam patterns of 10-element conventional array ( $B_p$ ), vector-sensor modulation term ( $B_v$ ) and the combined beam pattern ( $B_{comb}$ ) steered to  $\frac{\pi}{2}$ . This is a modified version of the plot shown in [13]

arrival is exactly  $180^\circ$  from the steering angle and nulled perfectly by the vector sensor modulation term. If the steering angle is not the broadside of the array the nulling by the vector sensor is still effective but does not exactly null the ambiguous arrival. The cases where the array is steered to  $\frac{\pi}{3}$ ,  $\frac{\pi}{4}$  and  $\frac{\pi}{6}$  are shown in figure 2-5. You can see that as the steering angle moves away from the broadside of the array

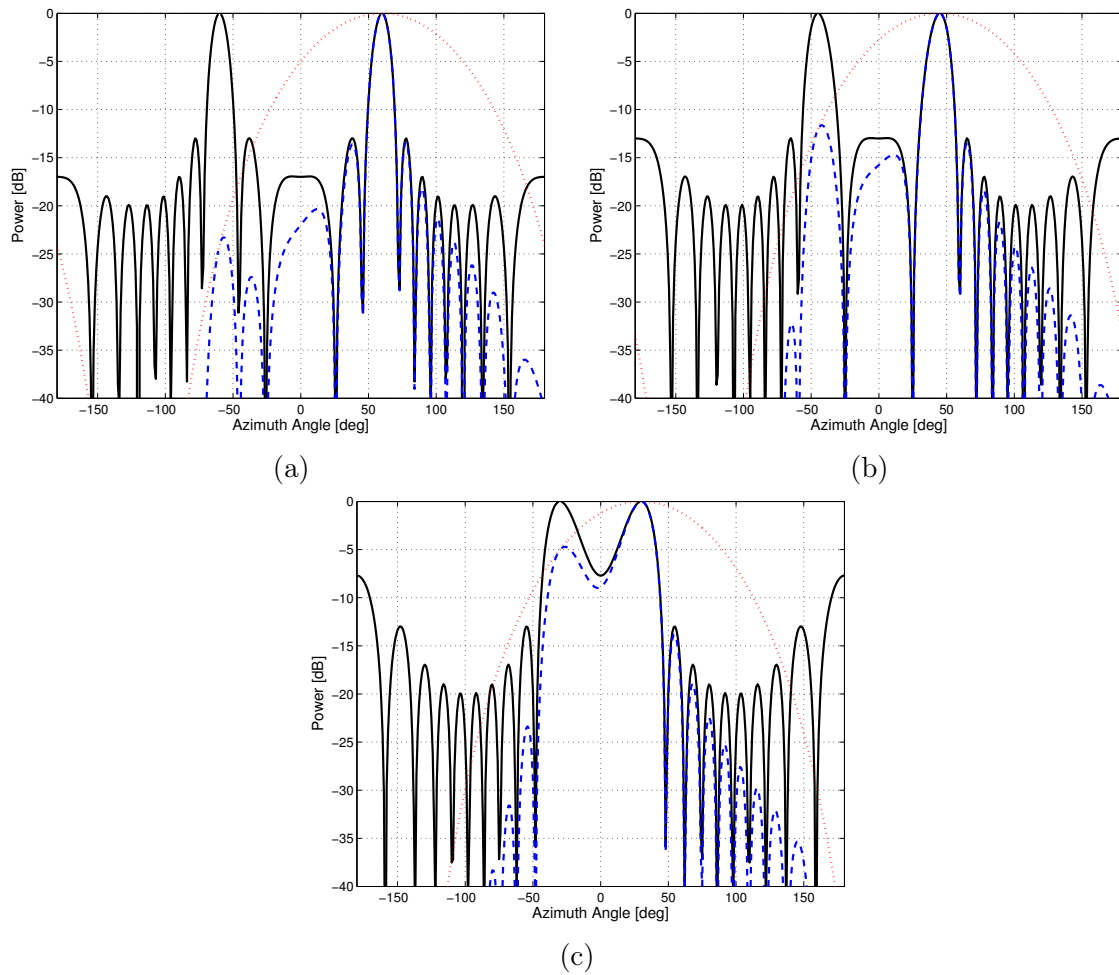


Figure 2-5: Beam patterns of 10-element conventional array ( $B_p$ ), vector-sensor modulation term ( $B_v$ ) and the combined beam pattern ( $B_{comb}$ ) being steered to  $\frac{\pi}{3}$  (a),  $\frac{\pi}{4}$  (b) and  $\frac{\pi}{6}$  (c). This is an expansion of the plots shown in [13]

the maximum power in the ambiguous peak increases. Even in case of steering to  $\frac{\pi}{6}$  as seen in figure 2-5(c) there is still nearly a 5 dB reduction from the peak power in the true arrival. The narrowest angle that the array can be steered and still result in a minimum of a 3 dB reduction from the peak power of the true arrival is approximately  $24.5^\circ$ .

The final advantage we want to discuss is the ability of vector sensors to discriminate grating lobes. The under sampling of a time-domain signal results in aliasing in the frequency domain [16] while spatially under sampling a plane wave will result in aliasing in the beampattern [27]. These spatial aliased versions of the signal are called grating lobes and result in a limitation in the use of a traditional pressure-sensor array above a specified design frequency. These grating lobes arise when the distance between elements of an array is greater than one-half the wavelength which is determined by the frequency. A more in depth discussion of this phenomena can be found in [13] and it was shown that the use of a vector sensor array will result in improved performance above the design frequency (where grating lobes begin to occur) of a corresponding pressure-sensor array. This performance comes in the form of nulling grating lobes due to their arrival at lower dB levels at the same design frequency.

### 2.2.3 Types of Vector Sensors

There are many different types of vector sensors that can be used for different applications. As discussed in the section on geometry this thesis deals specifically with a vector sensor that has three orthogonally oriented accelerometers plus an omnidirectional hydrophone collocated in the sensor housing. As discussed in [28] there are four common constructions of a vector sensors as in its most general sense a vector

sensor is composed of two or three spatially collocated but orthogonally oriented velocity sensors plus an optional collocated omni-directional hydrophone. This leads to three different configurations that we have not already discussed: three orthogonally oriented velocity sensors, two orthogonally and horizontally oriented velocity sensors and finally two orthogonally and horizontally oriented velocity sensors with a omni-directional pressure hydrophone. The construction of the array manifold for each of these is rather simple as you would just take the appropriate component parts from equation (2.3).

A vector sensor in general does not specify the kind of velocity measurement being performed however, this is an important detail. This is because there are two different ways to measure acoustic velocity: directly and indirectly. Direct measurement is measuring the acceleration that the incident pressure field is causing and then integrating using the appropriate relationship to determine the velocity. This can be accomplished using a piezoelectric crystal or also through use of a geophone which contains magnetic masses suspended by springs which induce a current in a wire coil which will yield the velocity.

Another way to measure acoustic velocity is indirectly through the use of closely spaced hydrophones and the use of the Navier-Stokes equations to derive an equation relating pressure and particle velocity. If we assume an inviscid fluid we can simplify the Navier-Stokes equations to the Euler equations.

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}^T \nabla \mathbf{v} = -\frac{\nabla p}{\rho} \quad (2.7)$$

where  $\mathbf{v}$  is particle velocity,  $\rho$  is density and  $p$  is pressure. As described in [13]

equation (2.7) can be reduced with the appropriate assumptions to

$$\mathbf{v} = -\frac{\mathbf{u}}{\rho c} p \quad (2.8)$$

where  $c$  is the sound speed and  $\mathbf{u}$  represents the unit vector pointing towards the source from the sensor which is merely the first three components in equation (2.3). This is important because as the directional hydrophones will be oriented with the coordinate axes you can determine the velocity via a linear relationship with the appropriate environmental data.

## 2.3 Adaptive Equalization

Adaptive equalization is extremely important in signal processing of acoustic communication signals due the nature of medium through which the signal is traveling. Since the ocean environment is continually changing, the equalizer must continually change. One method of modeling and accommodating the changes is via Doppler shifts and the time-update recursive least squares algorithm (TU-RLS) which we will describe later. An equalizer is a set of linear filters that compensates for intersymbol interference (ISI) and ambient noise in the received signal in a manner that allows us to recover the transmitted symbols. A computationally efficient method to accomplish this is the recursive least squares (RLS) algorithm which will be discussed below.

In a perfect world where we had full knowledge of the channel and no additive noise we would simply invert the channel response to recover the transmitted signal. Unfortunately in the real world this is not possible as inverting the the channel is not an appropriate solution as the inverse filter accentuates the frequencies exactly

where the signal power is lowest relative to the noise [17].

### 2.3.1 System Model

Before going any further we must have an understanding about the model of the system we are utilizing. A block diagram of this model is shown in figure 1-1. In this section we will go into detail about how each of the different blocks function and then in subsequent sections go into any further detail and provide derivations where they may be required.

#### The Channel

We define  $s[n]$  as the symbol that is transmitted at time  $n$ . The channel through which the symbol passes traveling from source to receiver introduces ISI. Additionally, there is random noise added to the signal that is represented at time  $n$  by  $v[n]$ . We designate  $x[n]$  as the signal received at the output of the channel at time  $n$ . The length of the channel response is defined as  $M$ . This clearly would imply that the received signal at time  $n$  depends on the transmitted signals  $s[n], s[n-1], \dots, s[n-(M-1)]$ . As we have discussed previously the channel response is varying so for now it will be defined at a given time  $n$  as  $\mathbf{h}[n]$  which gives us our received signal in the following equation:

$$x[n] = \mathbf{h}^H[n] \begin{bmatrix} s[n] & s[n-1] & \dots & s[n-(M-1)] \end{bmatrix}^T + v[n]. \quad (2.9)$$

It is implicit in equation (2.9) that we are treating the channel response as causal, this is a valid assumption because even if it were not we could introduce the proper delay to make it causal. Now to deal more clearly with the linear time-varying channel

response, at a given time  $n$  there are a defined set of channel coefficients the length of the channel response, these are represented by  $h_k[n], k = 0, 1, \dots, M - 1$ . This representation results in an output of the channel shown by the following convolution:

$$x[n] = \sum_{k=0}^{M-1} h_k^*[n]s[n-k] + v[n]. \quad (2.10)$$

This allows us to denote the channel coefficients at a given time  $n$  as a channel vector,  $\mathbf{h}[n]$ .

$$\mathbf{h}[n] = \begin{bmatrix} h_0[n] & h_1[n] & \dots & h_{M-1}[n] \end{bmatrix}^T \quad (2.11)$$

## Equalizer Filters

In a decision feedback equalizer (DFE) there are two filters: a feed-forward filter which is represented by  $\mathbf{g}_f$  with length  $N_f$  and the feedback filter represented by  $\mathbf{g}_b$  with length  $N_b$ . Put most simply the feed-forward filter balances attenuating noise and compensating for ISI and the feedback filter removes any remaining ISI. In slightly more detail the feed-forward filter inverts the channel estimate while at the same time ensuring that the noise introduced to the signal in the channel does cause the error to blow up. The feedback filter uses past decisions that have been made to eliminate any remaining ISI from the output of the feed-forward filter. The combined overall number of coefficients in the two filters is simply  $N = N_f + N_b$ .

The input to the feed-forward filter at a given time  $n$  is the vector of received signals from the last  $N_f$  samples at the channel output which we denote as  $\mathbf{x}[n]$ . The last  $N_f$  samples does not necessarily mean  $N_f$  symbols; in our case we are using a fractionally sampled signal so the number of symbols captured by the feed-forward



filter is actually  $\frac{N_f}{2}$ .

$$\mathbf{x}[n] = \begin{bmatrix} x[n] & x[n-1] & \dots & x[n-(N_f-1)] \end{bmatrix}^T \quad (2.12)$$

Similarly, the input to the feedback filter at a given time  $n$  are the decisions made at the previous  $N_b$  steps. We represent the decision made at time  $n$  by  $d[n]$  which is an estimate of the transmitted symbol at that specified time. Therefore we represent our input to the feedback filter by  $\mathbf{z}[n]$ .

$$\mathbf{z}[n] = \begin{bmatrix} d[n-1] & d[n-2] & \dots & d[n-N_b] \end{bmatrix}^T \quad (2.13)$$

As we are discussing an adaptive equalizer the filter coefficients are determined during each recursion using the decisions until the previous step ( $n-1$ ) and the input data all the way up to the current step ( $n$ ). However, when we update the filter coefficients at the end of each recursion they have utilized all of the data up to time  $n$  and are represented mathematically by  $\mathbf{g}_f[n|n]$  and  $\mathbf{g}_b[n|n]$ . When we step to the next recursion the feed-forward coefficients and feedback coefficients are then represented by  $\mathbf{g}_f[n|n-1]$  and  $\mathbf{g}_b[n|n-1]$ , respectively. This gives us an overall output of the equalizer represented by  $y[n]$ .

$$y[n] = \mathbf{g}_f^H[n|n-1]\mathbf{x}[n] - \mathbf{g}_b^H[n|n-1]\mathbf{z}[n] \quad (2.14)$$

This notation can be significantly simplified by creating combined filter coefficients and input vectors represented in equations (2.15) and (2.16), respectively.

$$\mathbf{g}[n|n-1] = \begin{bmatrix} \mathbf{g}_f[n|n-1] \\ -\mathbf{g}_b[n|n-1] \end{bmatrix} \quad (2.15)$$

$$\mathbf{u}[n] = \begin{bmatrix} \mathbf{x}[n] \\ \mathbf{z}[n] \end{bmatrix} \quad (2.16)$$

This then allows us to simplify equation (2.14) to

$$y[n] = \mathbf{g}^H[n|n-1]\mathbf{u}[n] \quad (2.17)$$

## Output and Decisions

The output represented in equation (2.17) is then input into a decision device where it is turned into an estimate of the symbol transmitted which is then subsequently used as an input to the feedback filter. You can have multiple stage equalizers where this decision would be input into another component of the system but in our case this decision device is a hard decision device where it will make its final approximation of the signal transmitted. It is important to stress that the output of the equalizer is only an estimate of the transmitted data, therefore the difference between the output of the equalizer and the input to the channel is the error.

### 2.3.2 Method of Least Squares

The method of least squares most simply determines the filter coefficients that minimize the squared-error between the channel input and the equalizer output. As Haykin describes in [6] this method is a model-dependent procedure to solve a linear filtering problem without making any assumptions about the statistics of the input to the filter. Looking at the equalizers we are discussing the inputs to the filter are the output of the acoustic propagation channel. This is important because it allows us to make no assumptions about the nature of the channel as we begin implementation of the equalizer. The channel coefficients act as the model to the ideal MMSE equalizer

filter coefficients and the total error between transmitted data (desired response) and the output of the equalizer is the summation of two different types of error.

The first part of the total error is in the model of the ideal coefficients. This is the error that would be achieved by the ideal MMSE equalizer and is called the minimal achievable error (MAE). The second part of the total error is the difference between the output of the least squares equalizer and the output of the ideal MMSE equalizer. This error is due to the error in our estimate of the ideal MMSE equalizer and is referred to as the excess error. The total error is represented as follows:

$$e[n] = e_{MAE}[n] + e_{excess}[n] \quad (2.18)$$

We have previously defined the desired response as  $d[n]$  therefore the total error is represented as follows:

$$e[n] = d[n] - y[n] \quad (2.19)$$

which can then be expanded using equation (2.14) to become:

$$e[n] = d[n] - \mathbf{g}^H[n|n-1]\mathbf{u}[n] \quad (2.20)$$

As previously mentioned we want to estimate the optimal filter coefficients to minimize the squared error over the interval from  $n_1$  to  $n_2$ . This is shown below:

$$\hat{\mathbf{g}} = \arg \min_g \sum_{i=n_1}^{n_2} |e[i]|^2 \quad (2.21)$$

where  $n_1$  and  $n_2$  represent the beginning and ending time indices of the interval over which we are optimizing the filter coefficients.

Combining equations (2.20) and (2.21) yields:

$$\hat{\mathbf{g}} = \arg \min_g \sum_{i=n_1}^{n_2} (d[i] - \mathbf{g}^H \mathbf{u}[i])(d[i] - \mathbf{g}^H \mathbf{u}[i])^* \quad (2.22)$$

If we then take the complex gradient [1] with respect to  $\mathbf{g}^H$  and then set it equal to zero it simplifies to the following equation:

$$\left( \sum_{i=n_1}^{n_2} \mathbf{u}[i] \mathbf{u}^H[i] \right) \mathbf{g} - \sum_{i=n_1}^{n_2} \mathbf{u}[i] d^*[i] = 0 \quad (2.23)$$

Let

$$\mathbf{\Phi} = \sum_{i=n_1}^{n_2} \mathbf{u}[i] \mathbf{u}^H[i] \quad (2.24)$$

where  $\mathbf{\Phi}$  is the  $M$  by  $M$  sample correlation matrix and

$$\mathbf{r} = \sum_{i=n_1}^{n_2} \mathbf{u}[i] d^*[i] \quad (2.25)$$

where  $\mathbf{r}$  is the sample cross-correlation vector between the desired response and the input. If we substitute equations (2.24) and (2.25) into (2.29) we are left with the matrix form of the normal equations for linear least-squares filters.

$$\hat{\mathbf{g}} = \mathbf{\Phi}^{-1} \mathbf{r} \quad (2.26)$$

Clearly you can see in equation (2.26) the  $\mathbf{\Phi}^{-1}$  term which automatically brings some concerns to mind. The first is that  $\mathbf{\Phi}$  cannot be a singular matrix or, in other words, it must be invertible. The other concern is that taking the inverse of a full rank matrix is a computationally expensive operation. The latter will be addressed

in the subsequent section while the former is addressed at initialization by diagonally loading the matrix as follows:

$$\Phi = \sum_{i=n_1}^{n_2} \mathbf{u}[i] \mathbf{u}^H[i] + \delta \mathbf{I}_M \quad (2.27)$$

Diagonal loading is discussed in much more detail in [6].

We have showed that you can determine the optimal filter weights; the model in this form does not however account for variability in the channel. We can minimize the effect of this channel variability by properly weighting each time step in forming the sample correlation matrix and the sample cross-correlation vector. We do this by applying an exponential window following a similar approach to the one shown above. The exponential least squares criterion is shown in equation (2.28). This will weight recent observations more heavily than past observations and in doing so helps to accommodate the time variability of the channel.

$$\hat{\mathbf{g}} = \arg \min_g \sum_{i=n_1}^{n_2} \lambda^{n_2-i} (d[i] - \mathbf{g}^H \mathbf{u}[i]) (d[i] - \mathbf{g}^H \mathbf{u}[i])^* \quad (2.28)$$

where  $\lambda$ , such that  $0 < \lambda < 1$ , is the forgetting factor that provides the rate at which the exponential window decays away. It is clear that a  $\lambda$  value closer to unity would be appropriate for a slowly varying channel and a  $\lambda$  value closer to zero would be more appropriate for a very quickly varying channel.

As we did previously taking the complex gradient [1] of the exponential least squares criterion with respect to  $\mathbf{g}^H$  and then setting it equal to zero simplifies to

the following expression:

$$\left( \sum_{i=n_1}^{n_2} \lambda^{n_2-i} \mathbf{u}[i] \mathbf{u}^H[i] \right) \mathbf{g} - \sum_{i=n_1}^{n_2} \lambda^{n_2-i} \mathbf{u}[i] d^*[i] = 0 \quad (2.29)$$

This then leads directly to the normal equations for exponentially windowed linear least-squares filters.

$$\Phi[n] = \sum_{i=n_1}^{n_2} \lambda^{n_2-i} \mathbf{u}[i] \mathbf{u}^H[i] \quad (2.30)$$

$$\mathbf{r}[n] = \sum_{i=n_1}^{n_2} \lambda^{n_2-i} \mathbf{u}[i] d^*[i] \quad (2.31)$$

In this section we have derived a very general least-squares solution equalization problem. We are able to determine the optimal filter weights for any specific segment of time. However, the channel in underwater acoustics is time-varying therefore even over a specified segment of time you are optimizing the determined filter coefficients there will be additional error due to this time-variability. We have also implemented an exponential window to weight the most recent observations more heavily than past observations. Finally, we have mentioned the computational expense of this basic least-square method due to the inversion of the sample correlation matrix.

### 2.3.3 Recursive Least Squares Algorithm

We can derive a recursive form of the least squares algorithm by noting that we can represent the sample correlation matrix and sample cross-correlation vectors as:

$$\Phi[n] = \lambda \Phi[n-1] + \mathbf{u}[n] \mathbf{u}^H[n] \quad (2.32)$$

$$\mathbf{r}[n] = \lambda \mathbf{r}[n-1] + \mathbf{u}[n] d^*[n] \quad (2.33)$$

As you can see both equations will require an initialization at the commencement of the algorithm but that will be discussed later.

Now to the issue of computational cost. As we mentioned in the previous section the inversion of  $\Phi$  is an expensive operation that is dependent on the length of the filter weight vector ( $M$ ). To invert  $\Phi$  which we previously defined as an  $M$ -by- $M$  matrix requires  $O(M^3)$  operations. If we implement a basic matrix algebra result known as the *matrix inversion lemma* or *Woodbury's identity* it can reduce the number of operations to  $O(M^2)$  [6]. If  $\mathbf{A}$  and  $\mathbf{B}$  are defined as two positive-definite  $M$ -by- $M$  matrices,  $\mathbf{C}$  is a  $M$ -by- $N$  matrix and  $\mathbf{D}$  is a positive-definite  $N$ -by- $N$  matrix then the following expressions define the matrix inversion lemma [6]. If:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H \quad (2.34)$$

then:

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \quad (2.35)$$

By applying the following relations a computationally efficient recursive least squares algorithm can be derived. The reader is referenced to chapter 10 of [6] for the full and detailed derivation.

$$\begin{aligned} \mathbf{A} &= \Phi[n]; \\ \mathbf{B}^{-1} &= \lambda\Phi[n-1]; \\ \mathbf{C} &= \mathbf{u}[n]; \\ \mathbf{D} &= 1; \end{aligned} \quad (2.36)$$

Table 2.2 shows the complete recursive least squares algorithm as it is derived in

[6, 20]. For notational simplicity  $\mathbf{P}[n]$  is defined as  $\Phi^{-1}[n]$  and Haykin has defined  $\hat{\mathbf{w}}$  as the filter coefficients that I have defined (and kept) as  $\hat{\mathbf{g}}$ .  $\mathbf{k}[n]$  is defined as the gain vector whose purpose is to scale the updates to the filter coefficients ( $\hat{\mathbf{g}}$ ) and the inverse sample correlation matrix ( $\mathbf{P}[n]$ ). Additionally, the initialization parameters are included where  $\delta$  (which is used for the diagonal loading) is assigned a small positive value for a high SNR environment and a large positive value for a low SNR environment. The choice of these values is to ensure the best convergence performance of the RLS algorithm and is described in much further detail in §10.4 of [6].

<b>RLS Algorithm</b>	
Initialize:	
$\hat{\mathbf{g}}[0]$	$= \mathbf{0}$
$\mathbf{P}[0]$	$= \delta^{-1}\mathbf{I}$
For each iteration, $n = 1, 2, \dots$ compute:	
$\mathbf{\Pi}[n]$	$= \mathbf{P}[n-1]\mathbf{u}[n],$
$\mathbf{k}[n]$	$= \frac{\mathbf{\Pi}[n]}{\lambda + \mathbf{u}^H[n]\mathbf{\Pi}[n]},$
$e[n]$	$= d[n] - \hat{\mathbf{g}}^H[n-1]\mathbf{u}[n],$
$\hat{\mathbf{g}}[n]$	$= \hat{\mathbf{g}}[n-1] + \mathbf{k}[n]e^*[n],$
and	
$\mathbf{P}[n]$	$= \lambda^{-1}\mathbf{P}[n-1] - \lambda^{-1}\mathbf{k}[n]\mathbf{u}^H[n]\mathbf{P}[n-1].$

Table 2.2: Adapted Version of Haykin's RLS Algorithm

### 2.3.4 Time-Update Recursive Least Squares Algorithm

In the end the RLS algorithm is a computationally efficient method of implementing the solution to method of least squares. The averaging window that was applied is not exclusive to the RLS algorithm and can be done in a similar manner in the non-recursive approach. It is a weakness of the LS criterion which the RLS algorithm



uses to calculate the optimal filter coefficients that in channels with large Doppler spreads, the RLS algorithm can exhibit subpar performance [3]. This effect was researched and analyzed in [3] where they also proposed some modifications to the RLS algorithm to improve its ability to handle large Doppler spreads. Utilizing the work in [3] we implement an algorithm termed the *Time-Update Recursive Least Squares* (TU-RLS).

TU-RLS was the primary equalization algorithm that was used in the work completed in this thesis. As such it is necessary to cover the additional steps this approach requires and update the methodology of the RLS algorithm with that method. The TU-RLS algorithm contains four major sections: (1) time-update step, (2) filtering, (3) measurement updates and (4) Doppler compensation coefficient updates. The middle two sections are covered in exactly the same manner as those steps represented in table 2.2, the first and fourth sections include additional steps necessary to minimize the previously mentioned weakness of the RLS algorithm and perform the Doppler compensation.

The Doppler coefficient vector in a  $N$ -length vector is represented by  $\mathbf{f}[n]$  and initialized to a vector of ones. For purposes that will be clear shortly we will also use the diagonal matrix form of the Doppler coefficient vector represented by  $\mathbf{F}[n]$ . At the beginning of the time-update step we will use the Doppler compensation coefficients to apply the necessary compensation to filter coefficients calculated in the previous iteration.

$$\hat{\mathbf{g}}[n|n-1] = \mathbf{F}[n-1] \hat{\mathbf{g}}[n-1|n-1] \quad (2.37)$$

The correlation matrix time update corresponding to equation (2.37) is:

$$\mathbf{P}[n|n-1] = \mathbf{F}[n-1] \mathbf{P}[n-1|n-1] \mathbf{F}^H[n-1] \quad (2.38)$$

This matrix update is representative of the propagation of the error covariance time update in a noise free state-space system.

The actual Doppler compensation steps are performed using the Normalized LMS (NLMS) algorithm in the following manner. The first step for this is to compute the "effective input" for the NLMS algorithm as follows<sup>1</sup>:

$$\mathbf{u}_e[n] = \hat{\mathbf{g}}^*[n-1|n-1] \circ \mathbf{u}[n] \quad (2.39)$$

With the effective input you are then able to update the Doppler compensation vector using the NLMS algorithm with the specified step size,  $\mu$ .

$$\hat{\mathbf{f}}[n] = \mathbf{f}[n-1] + \frac{\mu \mathbf{u}_e[n] (d[n] - y[n])^*}{\|\mathbf{u}_e[n]\|^2} \quad (2.40)$$

Finally, the updated Doppler compensation vector is then normalized on an element by element basis before moving on to the next iteration, where  $f_i[n]$  is the  $i^{th}$  element in the Doppler compensation vector ( $\mathbf{f}[n]$ ) vector.

$$f_i[n] = \frac{\hat{f}_i[n]}{\max(1, |\hat{f}_i[n]|)} \quad (2.41)$$

The overall TU-RLS algorithm is shown in table 2.3. It is not necessary to perform

---

<sup>1</sup>This step occurs prior to the time-update steps of the algorithm in (2.37) even though it is a part of the NLMS algorithm for Doppler compensation. This step should always be the first step of each iteration, this can be clearly seen in table 2.3

the bottom four steps in that particular order. You can perform the two Doppler compensation steps either before or after the updates to the inverse sample correlation matrix and the filter coefficients.

The TU-RLS algorithm gives the user one more parameter that needs to be optimized; the user must choose the proper  $\lambda$  and  $\mu$ . These are not parameters that can be chosen universally but must be optimized for each individual sound channel. This is because there is no typical acoustic channel or noise characteristic. These parameters can be chosen by running the equalizer in *training mode*<sup>2</sup> across a range of values for both  $\lambda$  and  $\mu$  and then determining the combination which provides you with the lowest bit error rate (BER). The results for the experiment performed in this thesis will be covered in chapter 5.

When comparing the bit error rates of equalizers using the TU-RLS and RLS algorithm the advantages of the TU-RLS algorithm become clear. The first is that the minimum BER using the TU-RLS algorithm is lower than that from the RLS algorithm at all averaging window lengths which tells you that you are getting better performance from the TU-RLS algorithm overall. The second advantage that the TU-RLS algorithm provides is that the absolute minimum of the BER occurs with a longer averaging window (i.e. a  $\lambda$  value closer to unity). This is important because it means that the TU-RLS algorithm allows you to utilize the data you have for longer time period which is especially important in a rapidly changing channel environment. Utilizing the TU-RLS algorithm has the overall effect of shifting a curve of averaging window versus BER down and to the right.

---

<sup>2</sup>*Training mode* is when the desired symbol is the actual transmitted symbol. There are many cases where this is not always known but for the purposes of this thesis, our equalizer was always run in training mode.

---



---

**TU-RLS Algorithm**


---



---

Initialize:

$$\begin{aligned}\hat{\mathbf{g}}[0|0] &= \mathbf{0} \\ \mathbf{P}[0|0] &= \delta^{-1}\mathbf{I} \\ \mathbf{f}[0] &= \mathbf{1}\end{aligned}$$

For each iteration,  $n = 1, 2, \dots$  compute:

$$\begin{aligned}\mathbf{u}_e[n] &= \hat{\mathbf{g}}^*[n-1|n-1] \circ \mathbf{u}[n] \\ \hat{\mathbf{g}}[n|n-1] &= \mathbf{F}[n-1] \hat{\mathbf{g}}[n-1|n-1] \\ \mathbf{P}[n|n-1] &= \mathbf{F}[n-1] \mathbf{P}[n-1|n-1] \mathbf{F}^H[n-1] \\ \mathbf{\Pi}[n] &= \mathbf{P}[n|n-1] \mathbf{u}[n], \\ \mathbf{k}[n] &= \frac{\mathbf{\Pi}[n]}{\lambda + \mathbf{u}^H[n] \mathbf{\Pi}[n]}, \\ e[n] &= d[n] - \hat{\mathbf{g}}^H[n|n-1] \mathbf{u}[n], \\ \hat{\mathbf{f}}[n] &= \mathbf{f}[n-1] + \frac{\mu \mathbf{u}_e[n] e^*[n]}{\|\mathbf{u}_e[n]\|^2},\end{aligned}$$

$$f_i[n] = \frac{\hat{f}_i[n]}{\max(1, |\hat{f}_i[n]|)},$$

$$\hat{\mathbf{g}}[n|n] = \hat{\mathbf{g}}[n|n-1] + \mathbf{k}[n] e^*[n],$$

and

$$\mathbf{P}[n|n] = \lambda^{-1} \mathbf{P}[n|n-1] - \lambda^{-1} \mathbf{k}[n] \mathbf{u}^H[n] \mathbf{P}[n|n-1].$$


---

Table 2.3: Adapted Version of TU-RLS Algorithm

### 2.3.5 Multiple Channels

All of the descriptions of the equalizers discussed in this chapter assume there is only a single channel of input data. However, in many cases and particularly in this thesis we will be utilizing multiple channels of input data. The extensions of these algorithms to a multiple channel input system is quite trivial and will only affect the feed-forward portions of the equalizers. Equation (2.12) showed that  $\mathbf{x}[n]$  would be the overall input vector for one channel with multiple channels you would simply stack the input vectors for each channel on top of each other. The process to obtain the multiple channel input is show in (2.42).

$$\begin{aligned}\mathbf{x}_i[n] &= \begin{bmatrix} x_i[n] & x_i[n-1] & \dots & x_i[n-(N_f-1)] \end{bmatrix}^T \\ \mathbf{x}[n] &= \begin{bmatrix} \mathbf{x}_1[n] & \mathbf{x}_2[n] & \dots & \mathbf{x}_k[n] \end{bmatrix}\end{aligned}\tag{2.42}$$

where  $\mathbf{x}[n]$  is the multiple channel input vector,  $\mathbf{x}_i[n]$  is the input vector for each of the channels and there are  $k$  channels.

Since the input vector is being lengthened we also must lengthen the filter coefficient vectors. Where it was previously of length  $N_f$  the feed-forward filter coefficient vector is now of length  $k * N_f$ . In other words each channel has a vector of filter coefficients of length  $N_f$  and they are stacked on top of each other in the same manner as the input vector. Recall that the feedback portion of our filter coefficients depends only on the decisions that have been made at each previous recursion. Although you can have multiple input channels there is still only one decision made. Therefore, the length of the feedback coefficient vector will remain the same. The new overall filter

coefficient vector is represented in equation (2.43) and is of length  $N = k * N_f + N_b$ .

$$\mathbf{g}[n|n-1] = \begin{bmatrix} \mathbf{g}_{\mathbf{f}_1}[n|n-1] \\ \vdots \\ \mathbf{g}_{\mathbf{f}_k}[n|n-1] \\ -\mathbf{g}_{\mathbf{b}}[n|n-1] \end{bmatrix} \quad (2.43)$$

where  $\mathbf{g}_{\mathbf{f}_i}[n|n-1]$  is the filter coefficient vector for the  $i^{th}$  input channel.

These changes have no effect on the construction of the overall input vector shown in equation (2.16) nor the output in equation (2.17) and thus even though there will be larger computational costs due to the size of the sample covariance matrix there is no increased complexity in the actual structure of the equalizers.

# Chapter 3

## Modeling

Often times in order to be able to best interpret your results it is necessary to do some modeling to ensure that you understand the environment in which you are operating. This is especially true when you are working on something that has never been accomplished before; every bit of information recorded helps to better understand and explain the results that are being received. In the case of the experiment described in chapter 4 we want to understand how the transmitted signals travelled through the water to the receiver array. There are many different modeling approaches such as normal modes, wave number integration, finite approximation and others that are all described in [18]; for this thesis we choose to use ray-based model implemented using BELLHOP.

### 3.1 BELLHOP and Ray-based Models

BELLHOP is a ray-based model that predicts acoustic pressure fields in the ocean environment [18]. It is an incredibly robust interface in so far as it allows the user

to not only specify the environment but also the type of output that is desired. The outputs that are specifically utilized in this thesis are transmission loss, basic ray trace, arrival and eigenrays. Below we will go further into detail about the information that each of those simulations provides us and how we interpret that information to assist our analysis.

BELLHOP generally makes few assumptions without definition by the user but a brief overview of the underlying physics as well as the restrictions of the ray-based models is appropriate here. Ray-based models are based upon an extension of Snell's law of refraction and the idea that an incident wave gives rise to a reflected and transmitted wave as shown in figure 3-1. Equation (3.1) shows Snell's law is shown

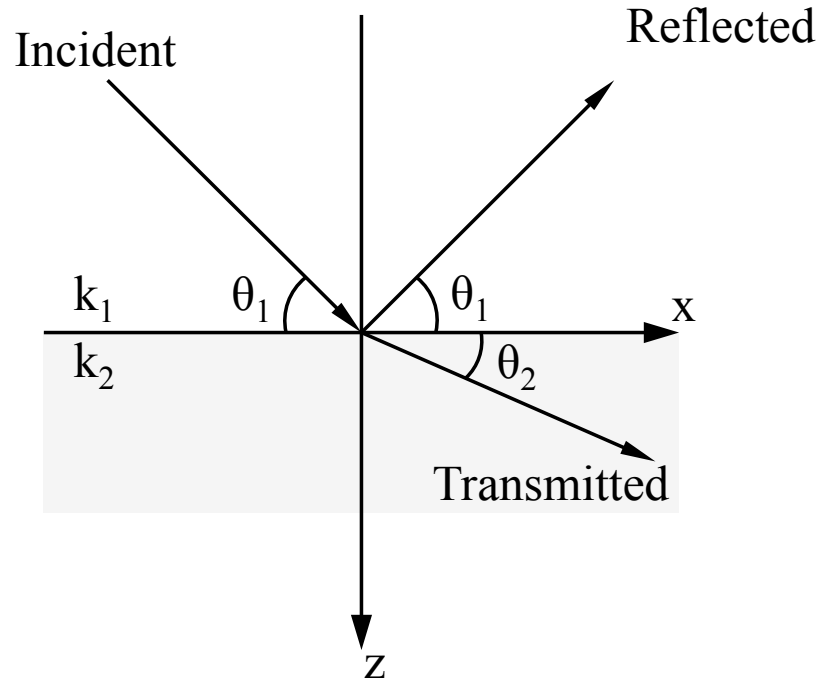


Figure 3-1: Reflection and transmission at a fluid-fluid interface

below in the acoustic sense where  $\theta_1$  is the incident angle,  $\theta_2$  is the transmitted angle,



$k_1$  and  $k_2$  are the wavenumbers of the first and second mediums, respectively.

$$k_1 \cos \theta_1 = k_2 \cos \theta_2 \quad (3.1)$$

The wavenumber at a given point in the water column is determined through the following relationship

$$k = \frac{2\pi f}{c} \quad (3.2)$$

where  $f$  is the frequency of the transmitted wave and  $c$  is the sound speed of the medium.

By modeling the ocean as a set of discrete layers stacked on top of each other; as shown in figure 3-2; where each boundary is a fluid-to-fluid interface. Therefore each transmitted wave becomes incident upon the layer beneath it. If you narrow the width of each layer until it's infinitesimally small, the stack of layers now approximates a fluid with a continuously varying wavenumber [11]. It is important to note that due to the fluid-to-fluid interface we are ignoring the effects of the reflected wave due to the weakness in the sound speed discontinuity which in this model approaches (and is treated as) zero. A more thorough description of this effect can be found in §1.6.1 of [11]. It can then be simply shown that the following relationship is observed between a ray starting point and any given receiver depth.

$$\theta_r = \arccos \left[ \frac{k(z_0)}{k(z_r)} \cos \theta_0 \right] \quad (3.3)$$

where the ray starting point is defined by a starting depth ( $z_0$ ) and launch angle ( $\theta_0$ ) and the angle of the ray ( $\theta_r$ ) at a given receiver depth ( $z_r$ ) is desired. It is important to note that at both the bottom and surface boundary the interaction is treated as a reflection controlled by the parameters of the surface or bottom layer. In the case

of the surface layer we treat it as a vacuum and thus have a perfect reflection. The bottom layer will be discussed in more detail in the next section.

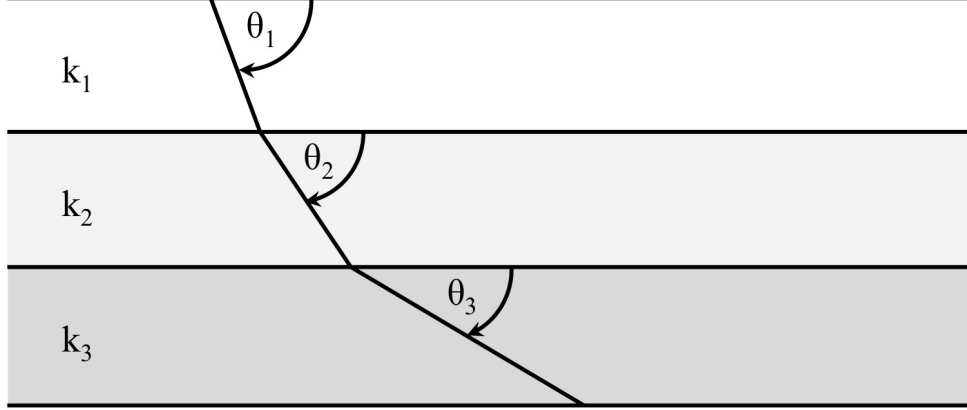


Figure 3-2: Ray refracting through a stack of fluid-fluid interfaces

This gives a very basic description of the physics behind how a ray-based model functions. In actuality it is more complicated than this and involves a derivation from the Helmholtz equation to determine the eikonal and transport equations. These equations can then be parameterized to give a set of first-order equations to describe the propagation of each ray and the transformations required for boundary interactions. The formation and the parameterization of the eikonal and transport equations represents a high frequency approximation of the solution to the acoustic wave equation [11]. The basic equations that govern the ray trace methods are shown below. The full mathematical derivation and description of ray-based models can be found in chapter 3 of [11].

$$\frac{dr}{ds} = c \xi(s), \quad \frac{d\xi}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial r} \quad (3.4)$$

$$\frac{dz}{ds} = c \zeta(s), \quad \frac{d\zeta}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial z} \quad (3.5)$$

along with the necessary initial conditions:

$$r = r_0, \quad \xi = \frac{\cos \theta_0}{c(0)} \quad (3.6)$$

$$z = z_0, \quad \zeta = \frac{\sin \theta_0}{c(0)} \quad (3.7)$$

where  $r$  and  $z$  represent the ray coordinates in cylindrical coordinates and  $s$  is the arc length along the ray.  $\zeta(s)$  and  $\xi(s)$  combined with the sound speed  $c$  represent the tangent vector along the ray. The initial conditions utilize the same ray starting point defined by the starting depth  $z_0$  and launch angle  $\theta_0$  described above.

The general idea of the ray-based models is to break the sound speed down into a nearly infinite number of layers based on an interpolation of the sound speed profile given in the user-defined environment file. BELLHOP specifically has a number of different types of interpolations that can be used: (1) C-linear, (2) N<sup>2</sup>-linear, (3) cubic spline, (4) analytic, and (5) quadratic approximation. Using this interpolated sound speed BELLHOP will trace each ray over a given range of launch angles to determine the ray's path through the ocean using the equations discussed above. BELLHOP treats each source by default as an omni-directional source which for this particular experiment it was. It is therefore up to the user to determine the number of launch angles for each run as a tradeoff between computational expense of the run and the results desired.

In more simple terms the user defines the sound speed at discrete depths and BELLHOP then interpolates between those depths to create a continuous sound speed profile. Using the interpolated sound speed profile, BELLHOP models the environment as a large number of infinitesimally small layers that approximates a

continuous sound speed profile. Using the interpolated sound speed BELLHOP will calculate the proper direction to travel using the equations shown above as well as the proper step size to take. The step size is determined based upon the total water depth, generally this is approximately a tenth of the water depth. However, this step-size is varied dynamically throughout the run to ensure that the ray will land on each discretely defined point of the sound speed profile.

One of the limitations of ray-based modeling is that as the range of the receiver increases the effect that slight environmental modeling errors have on the results compounds significantly. Consider a situation in which the sound speed profile has one erroneous value that is lower than it truly was. This lower value would then cause the ray to bend differently than it would normally in the ocean. The further the ray travels the more of an effect that this error would have compounding the mismatch between the actual path traveled and the modeled path. In our specific case due to the short range that the transmissions traveled to reach the receiver this is not considered an issue. An additional limitation to ray-based modeling is that the high frequency approximation inherent in the formation of the eikonal and transport equations leads to coarse accuracy in the results [11]. This is acceptable in our situation because we are only attempting to get a basic understanding of the experimental environment not a perfect representation.

## 3.2 Inputs

As we have mentioned above the environment file is the input the BELLHOP programs uses to determine the necessary parameters to run the desired rays. The most important part of this input is the sound speed profile which was determined as a part of the experiment described in chapter 4. It is important that we are using a

sound speed profile that was not perfectly in-situ for the vector sensor portion of the experiment, this will be described in more detail in section 4.2. Figure 3-3 shows the variation of the sound speed profile over the range of a few days. Based on the time of day that the vector sensor transmissions were performed we have isolated an approximate sound speed profile to use for our analysis, shown in figure 3-4.

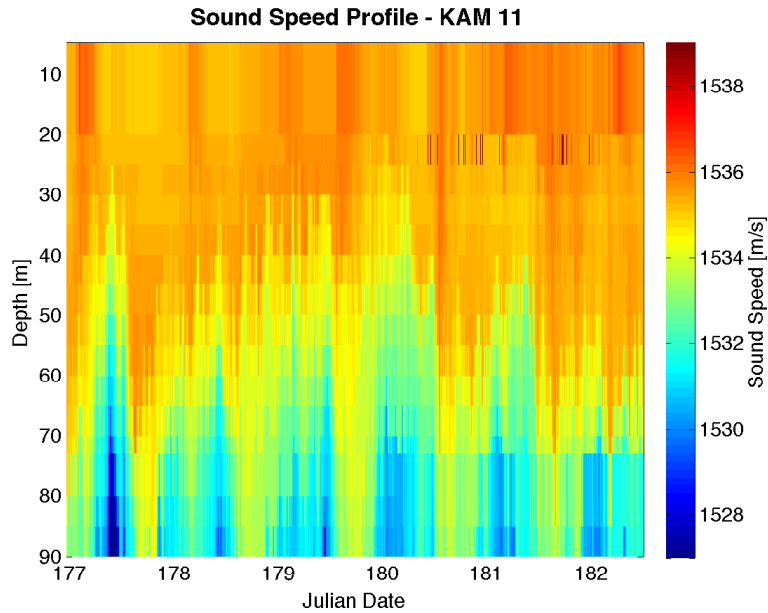


Figure 3-3: Sound speed profile of KAM11 from JD 176-182

You will note that this sound speed profile only goes down to a depth of 90 meters due to the shallow water nature of the other experiments being performed during KAM11. Due to the nature of the range that was being used for the experiment and to maximize the use of that range the vector sensor experiments were performed off of the range in deeper water than the sound speed profiler was able to characterize.

We discussed above that there were different options to use for the interpolation for the sound speed profile. The spline fit was chosen since we can tell from figure 3-4 that the sound speed seems to vary relatively smoothly. C-linear was not chosen

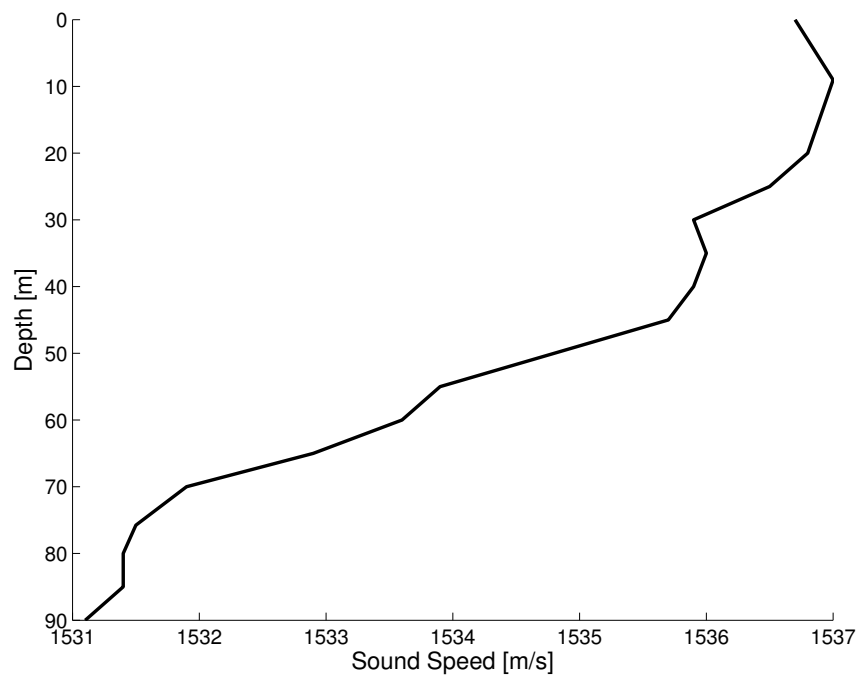


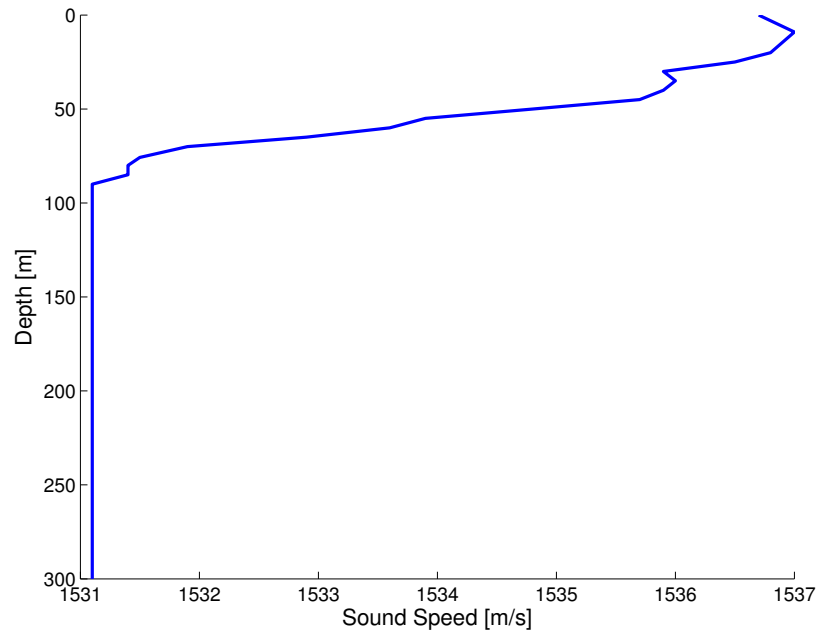
Figure 3-4: Sound speed profile used for BELLHOP simulations

because of the changes in slope below 10 meters. The  $N^2$  linear interpolation was not chosen because from [18, 11] we know that in cases where the sound speed varies smoothly that the spline fit provides smoother looking ray traces than a  $N^2$  linear fit.

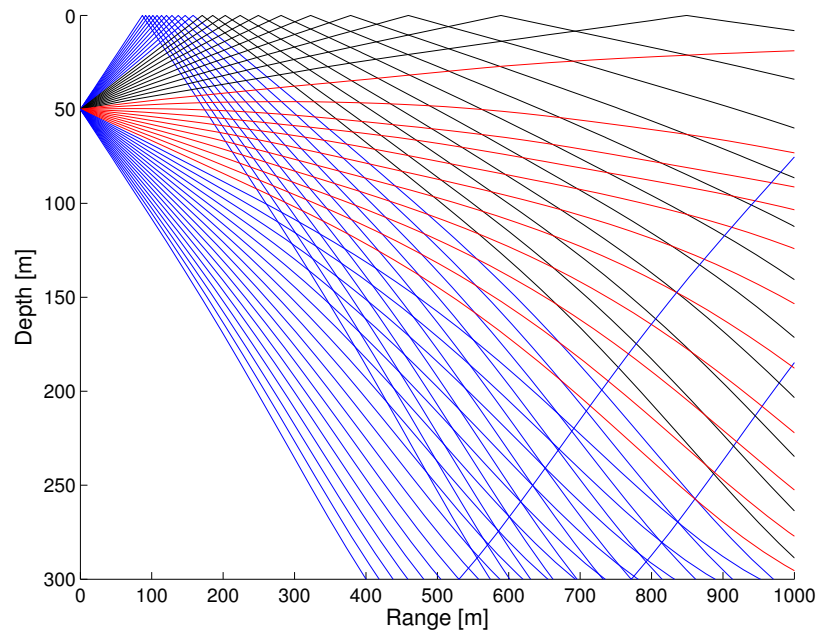
We also input the center frequency of the acoustic communications transmissions which was 3.8kHz. The rays are considered frequency independent because the wavenumber only varies during each simulation as a function of sound speed (see equation (3.2)). However, the frequency has an effect on the ray step size, since BELLHOP makes the assumption that at higher frequencies more accurate ray trajectories are required [18] and thus shorter step sizes.

As far as the treatment of the boundaries we treated the surface boundary as an air-water interface which in BELLHOP is modeled as a vacuum. The input for the bottom boundary is of no relevance in our case because there are no bottom bounce interactions in the range we are concerned with. However, to verify this assumption the bottom was modeled as a hard bottom that would give a reflection very similar to that of a perfect reflector. This was accomplished in our modeling by maintaining an isovelocity sound speed from 90 meters to the bottom sounder depth of 300 meters. The results from this showed that once a ray went deeper than 90 meters it would not turn back and be received at the vector sensor array through either refraction or a bottom bounce. This ray trace as well as the estimated sound speed profile can be seen in figure 3-5 where the red, black and blue lines represent direct path, surface bounce and bottom bounce arrivals, respectively. You can see that the bottom bounce arrival that comes shallowest is at a depth of around 50 meters at the range of the vector sensor array.

In addition to the parameters that have been discussed above the user is also required to input the source depth as well as the range and depths of the receivers.



(a)



(b)

Figure 3-5: (a) Estimated sound speed profile (b) Ray trace of estimated sound speed profile



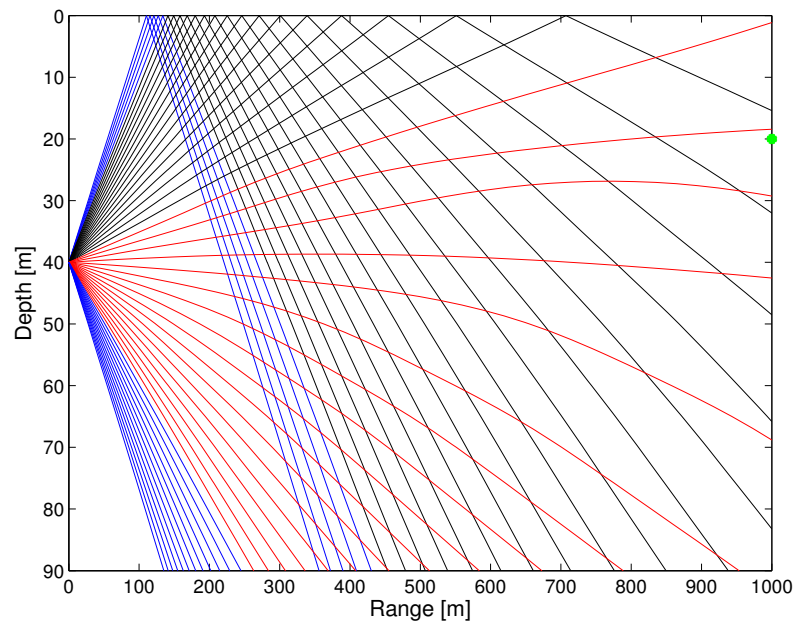
For our experiment the source was deployed at depths of both 40 and 50 m. The vector sensor array was a vertical array deployed to a depth of the 20 m with 33 cm inter-element spacing. However, as will be discussed further in chapter 4 two of the sensors did not record good data so we analyzed only for the functioning vector sensor elements at 20 m and 20.99 m.

Finally, the last part of the input is the type of trial for BELLHOP to perform. As mentioned above for our analysis we had BELLHOP perform transmission loss, ray trace, eigenray and arrival simulations. Exactly what the purpose of each of these runs will be discussed in §3.3.

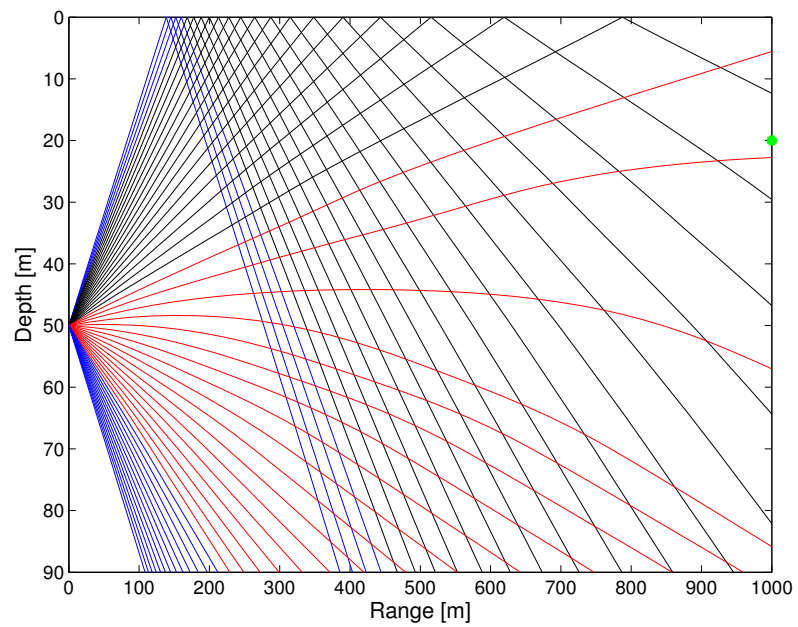
### 3.3 Results and Interpretation

The first trial that was performed was the basic ray trace option which performed a basic ray trace from the source to the location of the receiver. Fans of many different launch angles were performed but the displayed figure was generated to show all of the basic paths but also so that the results can be clearly interpreted. You can see from figure 3-6 that we have the bottom bounce rays, shown in blue, which travel deeper than 90 meters and do not return above that depth at the ranges with which we are concerned. These deep rays include both the rays that go directly into the deep sound channel as well those that hit the surface and then subsequently perform a bottom bounce. Next there are the surface bounce rays, shown in black, these are the rays that are transmitted with launch angles towards the surface where they are reflected back into the water column where they can interact with the receiver. Finally there are the waves that are launched at angles nearer to zero and travel via a direct path to the receiver, these are shown in red.

The eigenray trial is essentially a subset of the basic ray trace option in that



(a)



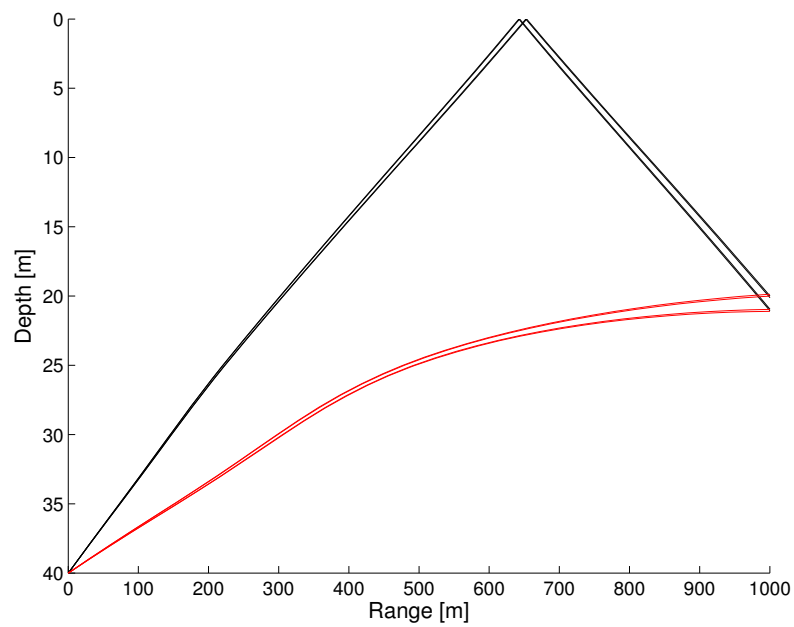
(b)

Figure 3-6: Ray traces for (a) source at 40 meters and (b) source at 50 meters

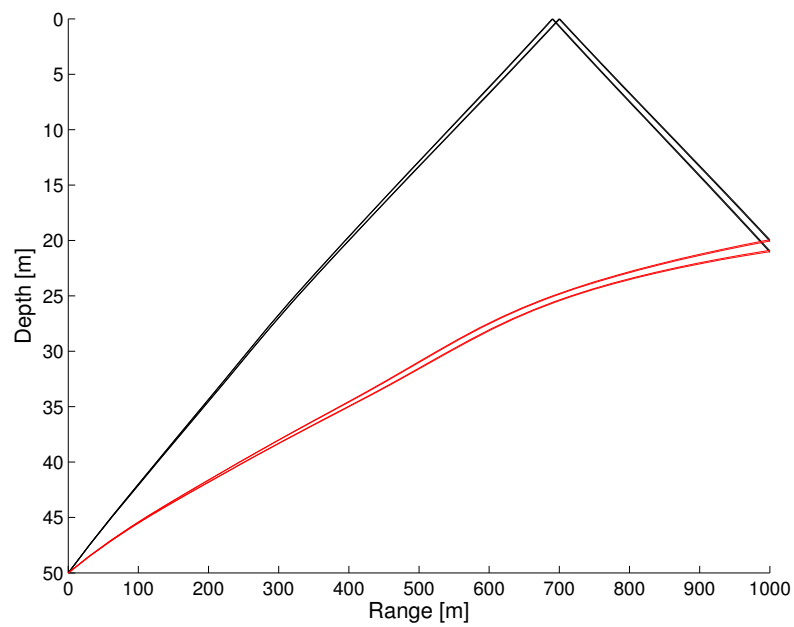
windows the fan of launch angles to the angles that result in a ray that brackets the specified receiver location. The results from this are shown in figure 3-7 where the black rays are the surface bounce rays and the red rays are the direct path. From this figure as well as figure 3-5 you can see that our assumption that the deep rays would not refract back up to the receivers or bounce off the bottom to our receivers was validated. The interesting thing that can be seen from these two plots are that the angles of arrival at the receivers are near horizontal. Based upon these figures and our processing of the vector sensors we have found that the z-axis accelerometer does not provide useful data for equalization in this case. This is an area for further study in the future where the range from source to receiver is distant enough where you are getting arrivals from the deep rays at steep angles where that information would be much more useful.

Additionally these figures tell us that we should be seeing two main arrivals after the transmission, the direct path and the surface bounce. This is additionally confirmed by the arrival time and amplitude trial. This trial shows that there are two arrivals at each of the vector sensor elements. It also shows the separation in the arrivals is 1.1 msec with a 50 m source depth and 0.87 msec with a 40 m source depth. This makes intuitive sense as well as we would expect the 40 m source depth surface bounce to have a shorter distance to travel relative to the 50 m source depth as it has a shorter distance to travel to reach the surface. The results of this trial for the 50 m source are shown in figure 3-8. The amplitudes and arrival times for both sources are nearly identical and have not distinctive difference thus only the figure for the 50 m source is shown.

The final trial that was run was the transmission loss trial. This trial determines the intensity of the transmission as it travels through the water column with respect to a source of unit strength. The results from this trial are shown in figure 3-9. You



(a)



(b)

Figure 3-7: Eigenray traces for (a) source at 40 meters and (b) source at 50 meters

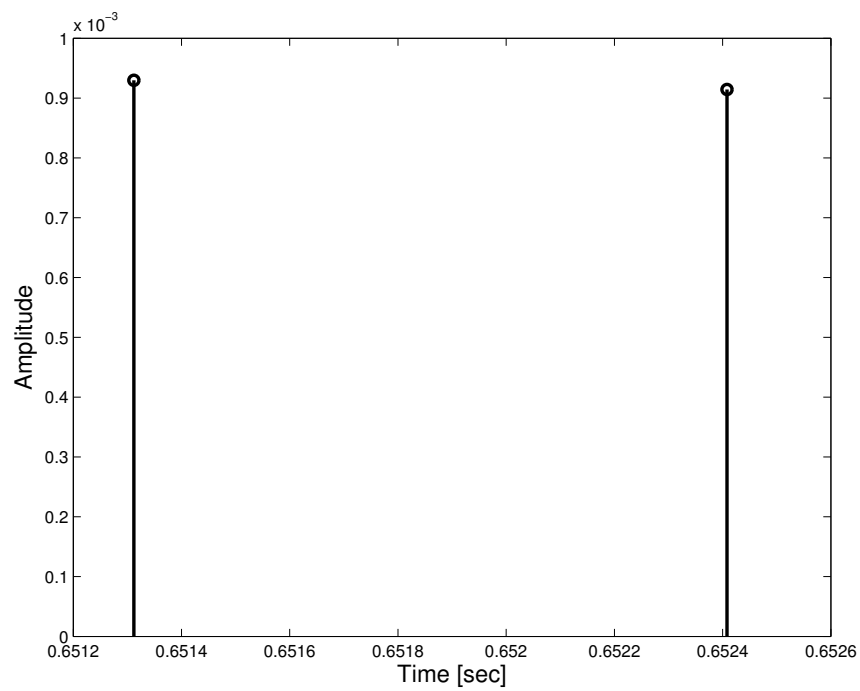


Figure 3-8: Arrivals with respect to time at the array for a 50 m source

can see from this that the arrivals coming from both the direct path and the surface bounce have a similar intensity or signal strength. We should expect this to show up when looking at the impulse response of the channel and the different arrivals later on in section 4.3.

You can see from these results that modeling the environment that the experiment is performed in can give us significant insight into the results that we have received or expect to receive. Ray-based models are an accurate and computationally inexpensive model that we have utilized to provide us with that insight.

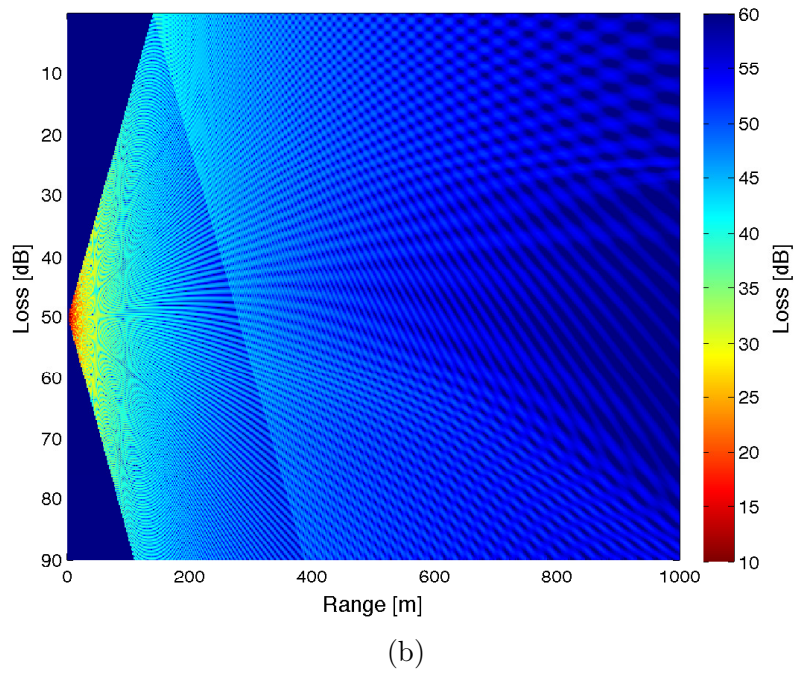
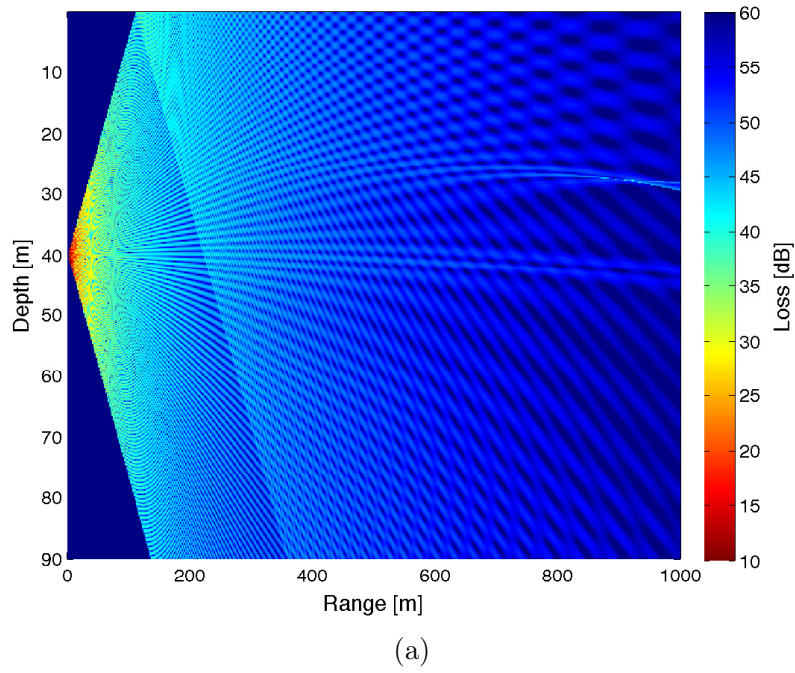


Figure 3-9: Tranmission loss plots for (a) source at 40 meters and (b) source at 50 meters





# Chapter 4

## The Experiment

### 4.1 Introduction

In July 2011 the Kaua'i Acoustic MURI 2011 (KAM11) experiment was conducted off the western coast of Kaua'i, Hawai'i at the U.S. Navy's Pacific Missile Research Facility (PMRF). The objective of this experiment was to advance the goal of coupling oceanography, acoustics and underwater communications by obtaining appropriate acoustic and environmental data [9]. For full details on the experiment the reader is referenced to the associated trip report, log book and subsequent paper [9, 10, 8]. This chapter will only cover the portions of the experiment that were associated with generating the vector sensor data utilized in this thesis.

Section 4.2 discusses the conditions and locations of the source and receiver. We look at the time variability of the channel in section 4.3. The nature of the transmitted data sequence and how it was constructed is discussed in section 4.4. Finally, the additive noise used to degrade the received signals as well as the process to do so is reviewed in section 4.5.

## 4.2 Physical Geometry

The data used in this experiment was collected during a deployment that occurred on 25-26 June (JD176-177). The location where the data was collected with the vector sensor array (VSA) was south of the area in which most of the remainder of the experiment was conducted. As a result, the thermistor array used to characterize the environment was approximately 20 km from the location of the VSA and can be considered to give only approximate environmental information [10].

The VSA itself was a 4-element (4-channels/element) autonomous vector sensor array configured as a vertical line array with 33 cm inter-element spacing ( $\lambda/2 \approx 2.2\text{kHz}$ ) at a depth of 20 m. Each of the elements was a Wilcoxon VS-205 with the numbering from the surface down (i.e. the sensor closest to the surface was 1 and the sensor closest to the bottom was 4). The mooring was freely drifting and decoupled from surface motion. The sampling rate of each channel was 45045 Hz. Each of the elements was a 4-channel vector sensor with one omni-directional hydrophone co-located with three orthogonally mounted piezoelectric crystal accelerometers.

During each VSA deployment the source described in [9] transmitted the signals for approximately two hours at a range of approximately 1 km from the array. The transmitted data sequences in the deployment were a rotation of three different sequences each lasting two minutes.

## 4.3 Time Variability

The conditions of the channel varied significantly over the course of the KAM11 experiment and even though the VSA deployment only lasted two hours the variability still had an impact. These variabilities are caused by a number of different factors

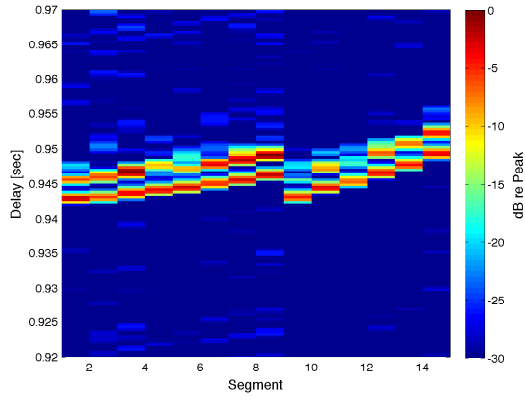
including changing wind and weather conditions along with tidal change and internal waves. Even though the VSA mooring was decoupled from the surface motion itself the decoupling was not perfect and some effects of surface motion on the array and on the surface reflected path are present in the data.

Figure 4-1 shows how the channels can change even over the short time frame of two minutes. These channel responses show not only the variability of the channel with time but also the two distinct arrivals from the direct path and surface bounce. The two different arrivals based on these channel impulse responses are separated by approximately 2 msec which compares favorably to those values determined through BELLHOP modeling in section 3.3. These channel impulse responses can also be used to ensure that your feed-forward filters in the equalizer are long enough which will be discussed further in section 5.4.

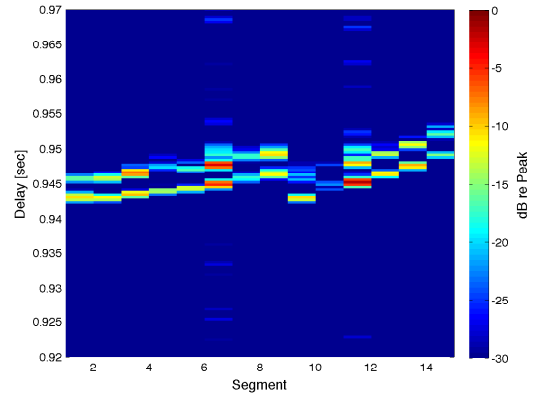
## 4.4 Data Structure

The transmitted data sequences utilized for this thesis were a sequence of 16 segments of signals. One full transmit sequence included two repetitions of the 16 segments with a 6 second silent period in between, each repetition had a total length (including signal and silent period) of 60 seconds. The spectrum of one repetition of the transmitted signal can be seen in figure 4-2.

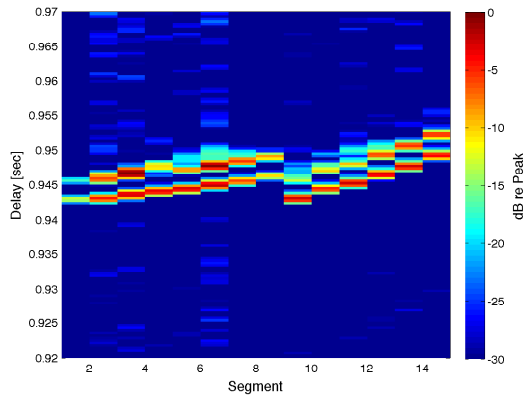
Each of the signals we utilized was a maximal length sequence (MLS) at the Nyquist rate, more commonly referred to as an m-sequence. The MLS form of the signal was chosen due to the autocorrelation properties that it exhibits under differing lag conditions; high autocorrelation under zero lag conditions and low autocorrelation under non-zero lag conditions. This property is best demonstrated when considering that as the length of the MLS increases towards infinity the autocorrelation functions



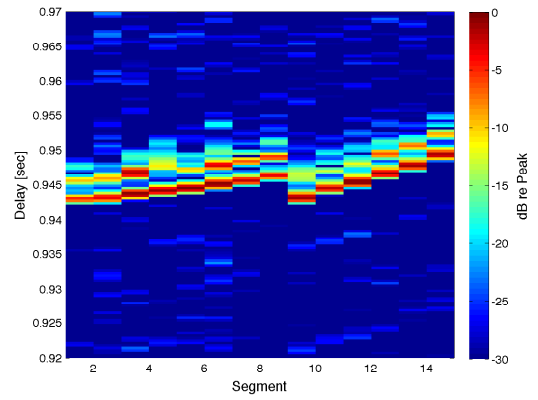
(a)



(b)



(c)



(d)

Figure 4-1: Normalized channel impulse response of the (a) pressure, (b) x-velocity, (c) z-velocity and (d) y-velocity elements of a single sensor of the vector sensor array.

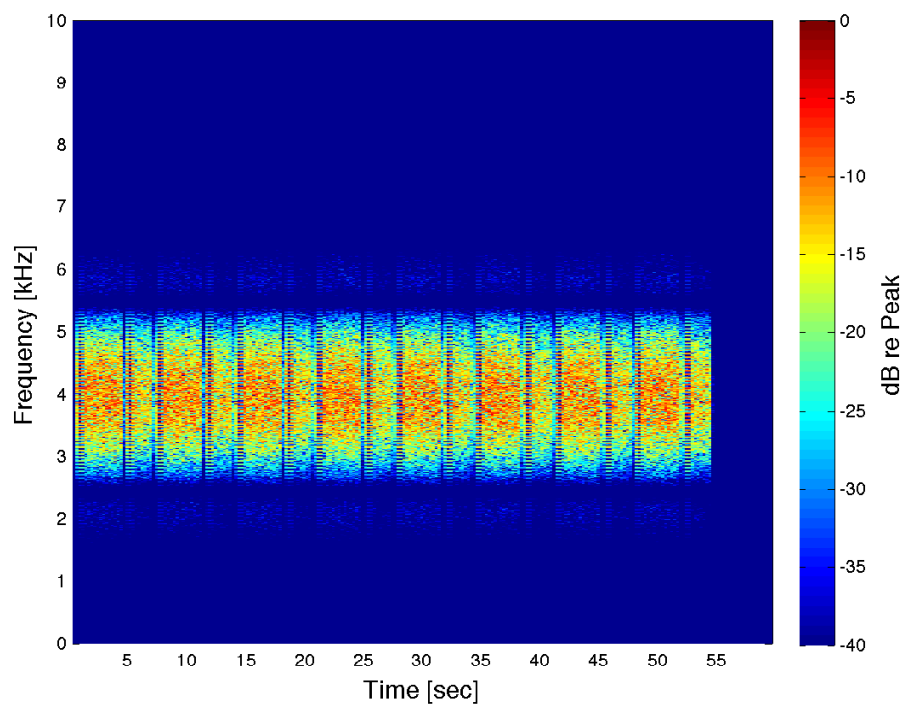


Figure 4-2: Spectrogram of one repetition of the transmitted signal

approaches the Kronecker delta function. A more detailed description of m-sequences and their properties can be found in [4].

Each of the transmitted segments was built from a combination of a detect data sequence, a training data sequence and a specified data segment (there were five different data segments used). The detect data sequence was formed from seven repetitions of a 127 symbol length detect m-sequence for a total detect data sequence length of 889 symbols. The training data sequence was an additional 417 symbols and finally the specified data segment was a 4000 symbol segment resulting in an overall transmitted segment 5306 symbols. The structure of these signals is also shown in table 4.1. Each of these symbols was modulated onto a Gaussian pulse

Sequence Name	Number of Symbols	Number of Repetitions	Total Number of Symbols
detect m-sequence	127	7	889
training data sequence	417	1	417
data segment	4000	1	4000
<b>Total:</b>			<b>5306</b>

Table 4.1: Structure of the transmitted signals

of length 72 and thus we transmitted 72 samples/symbols. The transmission signal was a Binary Phase Shift Keyed (BPSK) signal that was modulated to the carrier frequency of 3995 Hz that was sampled at 100,000 samples per second. The received signal at the VSA was recorded at 45045 samples per second and then resampled back to 100,000 samples per second.

## 4.5 Noise

The signal-to-noise ratio (SNR) of the different segments depend on the individual element of the sensor and were found to vary between 20-35 dB for the various segments. In order to fully analyze the capability of the vector sensors to be beamformed we needed to explore when the signal levels on each of the elements of the vector sensor were degraded to 20, 10 and 0 dB. We accomplished this by degrading the baseband received signal to the desired SNR by taking baseband received signals where there were no transmissions and scaling and adding them to the received signals from the times when there were transmissions. We define our scaling factor as  $\alpha$  and the equation below demonstrates mathematically how we degraded the SNR;

$$bbsig_{corr} = bbsig + \alpha bbnoise_{add} \quad (4.1)$$

where  $bbsig$ ,  $bbsig_{corr}$ ,  $bbnoise_{add}$  represent the original basebanded signal, the degraded basebanded signal and the basebanded additive noise segment.

The additive noise that was used was recorded by the vector sensor array as segment 2 of the deployment. This segment was found to have no signal present and no distinct sources in the frequency band that we are concerned with (2.5 - 5.5 kHz). This file similarly to the signal files as described in section 5.2 was basebanded, filtered, downsampled and then appropriately scaled to give the desired SNR. The spectrogram and energy in the desired band of 2.5-5.5 kHz from that file for all four channels is shown in figures 4-3 and 4-4, respectively. These figures show us that when this noise is added to the signal there are only a few small transients that have two orders of magnitude of energy less than that of the signal in the frequency band that the signal lies in. Additionally, the spectrogram shows that the

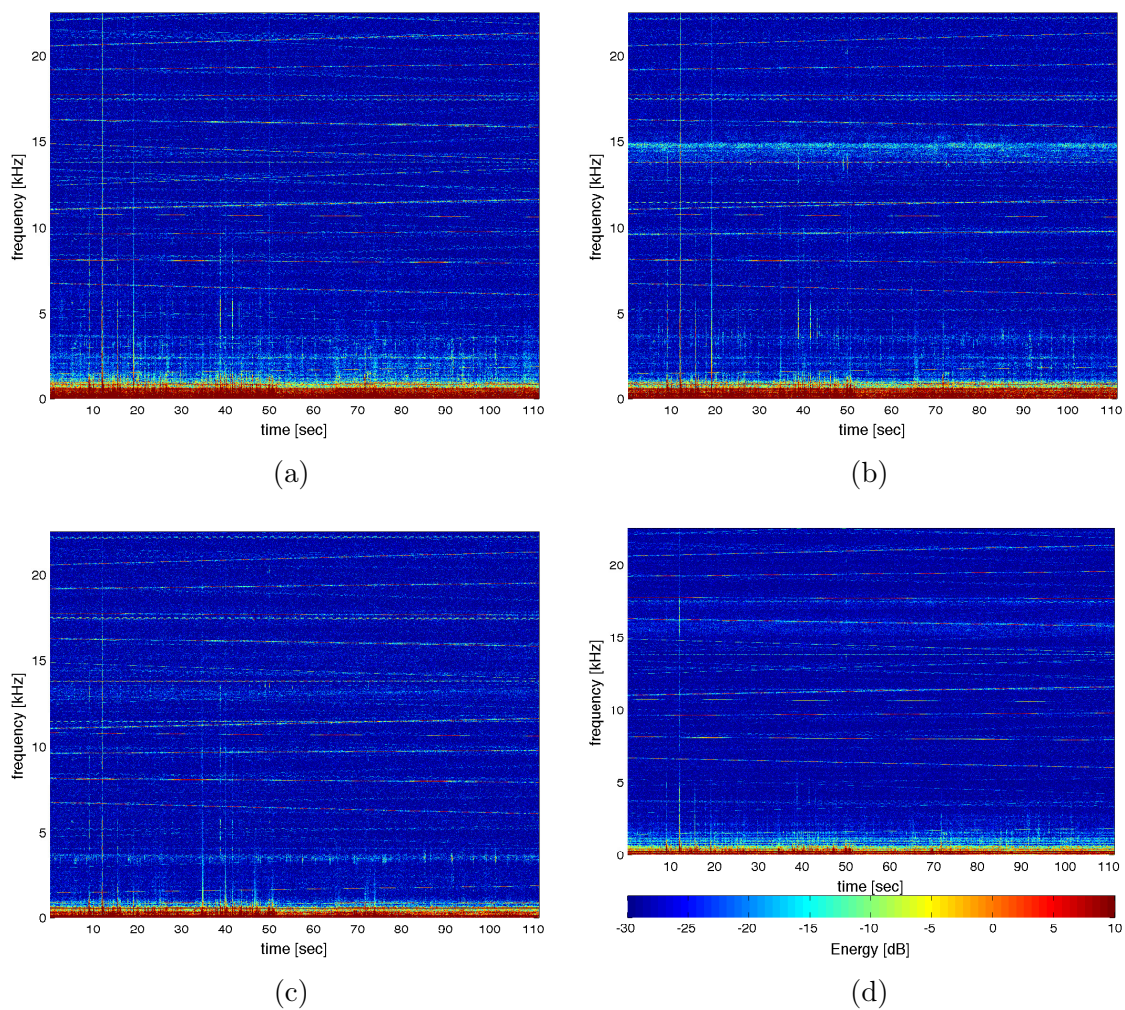


Figure 4-3: Spectrograms from segment 2 which was used as additive noise: (a) Pressure, (b) X-velocity, (c) Y-velocity and (d) Z-velocity



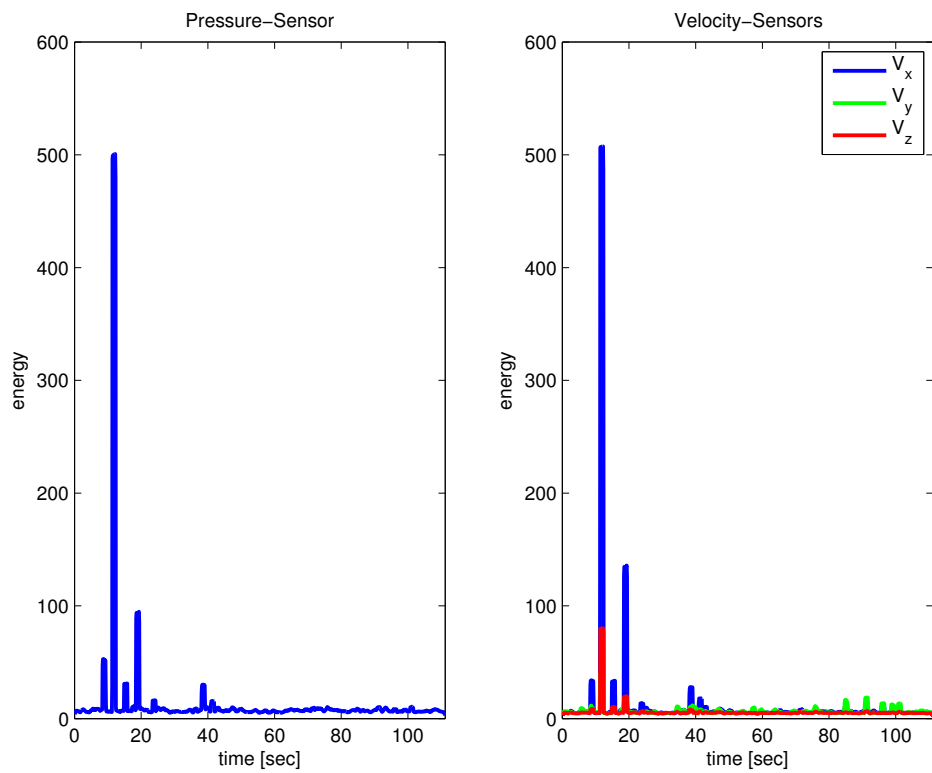


Figure 4-4: Energy of noise file in the frequency band of the signal

large concentration of the ambient noise is in the lower frequency band. Finally, the verification that no signal was present was done by performing a matched filter of the noise file with the detect data sequence which showed no peaks.

The scaling of the additive noise is not performed through just a simple addition of the noise. First the appropriate scaling factor must be determined and then the additive noise scaled before it is added to the received signal to give you the desired SNR (recall equation (4.1)). This derivation for determining the scaling factor ( $\alpha$ ) is shown below:

Initially the SNR of the received signal is shown in equation (4.2) and the SNR of the desired signal is seen in equation (4.3), where  $SNR$  and  $SNR_d$  are the native and desired SNR's respectively and  $\alpha$  is the additive noise scaling factor.  $\sigma_R^2$ ,  $\sigma_{N_1}^2$  and  $\sigma_{N_2}^2$  are the variance of the signal plus noise, noise alone but in the same segment as the signal plus noise portion and the noise alone segment, respectively.

$$SNR = 10 \log_{10} \left( \frac{\sigma_R^2 - \sigma_{N_1}^2}{\sigma_{N_1}^2} \right) \quad (4.2)$$

$$SNR_d = 10 \log_{10} \left( \frac{\sigma_R^2 - \sigma_{N_1}^2}{\sigma_{N_1}^2 + \alpha^2 \sigma_{N_2}^2} \right) \quad (4.3)$$

By expanding the argument of equation (4.3) as shown in equation (4.4) and using basic logarithmic relationships you then arrive at equation (4.5).

$$SNR_d = 10 \log_{10} \left( \left( \frac{\sigma_R^2 - \sigma_{N_1}^2}{\sigma_{N_1}^2} \right) \left( \frac{1}{1 + \alpha^2 \frac{\sigma_{N_2}^2}{\sigma_{N_1}^2}} \right) \right) \quad (4.4)$$

$$SNR_d = 10 \log_{10} \left( \frac{\sigma_R^2 - \sigma_{N_1}^2}{\sigma_{N_1}^2} \right) + 10 \log_{10} \left( \frac{1}{1 + \alpha^2 \frac{\sigma_{N_2}^2}{\sigma_{N_1}^2}} \right) \quad (4.5)$$

You can clearly see that the first term is the native SNR then by using basic logarithmic relationships and expressions and then rearranging you are left with the following relationship in equation (4.6).

$$SNR - SNR_d = 10 \log_{10} \left( 1 + \alpha^2 \frac{\sigma_{N_2}^2}{\sigma_{N_1}^2} \right) \quad (4.6)$$

By then solving for  $\alpha$  you arrive at equation (4.7) that determines the additive noise scaling factor. This scaling factor is then used as shown above in equation (4.1) but redisplayed here to degrade the SNR of the basebanded signal to the desired level.

$$\alpha = \sqrt{\frac{\sigma_{N_1}^2}{\sigma_{N_2}^2} \left( 10^{\frac{SNR - SNR_d}{10}} - 1 \right)} \quad (4.7)$$

$$bbsig_{corr} = bbsig + \alpha bbnoise_{add}$$

It is important to note that degrading the signal files involves degrading each individual element of the sensor on its own. In the cases where the native SNR was already lower than the desired SNR that signal was left alone as there is no way to increase the signal level in the file.



# Chapter 5

## Algorithms and Processing

### 5.1 Introduction

In this thesis our goal was to utilize vector sensors to enable us to successfully process acoustic communication signals at a lower SNR by utilizing the inherent directionality of a vector sensor. This chapter will initially focus on our processing of the received acoustic communication signals discussed in chapter 4 and more specifically in section 4.4. We will then delve into the method that we used to utilize the directionality of the vector sensors to process the received signals at degraded SNRs. This thesis has only focused on the manner in which to beamform the received signals and then the subsequent equalization of these signals. All of the processing is post-processing in this thesis, real-time processing is an area for future research.

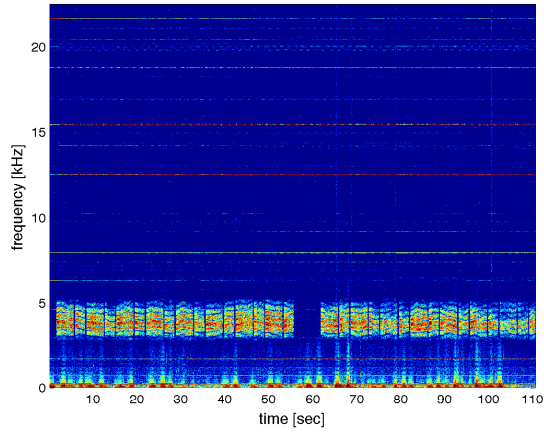
Section 5.2 will discuss the signal processing that was performed on the received signals from the vector sensor array in addition to the manner in which the signals were degraded to the appropriate levels. Section 5.3 will discuss the methodology of how the beamforming was performed as well as how it can be easily expanded

to three dimensions. In section 5.4 we discuss how the variable parameters in the adaptive equalizer were chosen. Finally in sections 5.5 we will discuss the results of the beamforming and adaptive equalization as well as the improvements that we saw and the computational savings.

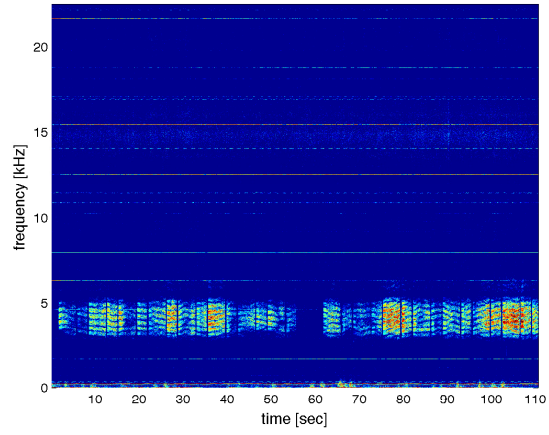
## 5.2 Received Signal Processing

The received signals from the vector sensor array (VSA) were recorded in 110 second segments. The spectrograms of every received signal segment were analyzed and we have chosen segment number 86, segment 101 and segment 104 from the VSA deployment as our data files for further analysis. The spectrum of the received signals for each segment can be seen in figures 5-1, 5-2 and 5-3. These figures include the spectrum from each of the components of one sensor from the vector sensor array: (1) pressure sensor (omni-directional hydrophone), (2) x-direction accelerometer, (3) y-direction accelerometer and (4) z-direction accelerometer. These spectrograms were determined with the dc component removed from the signal.

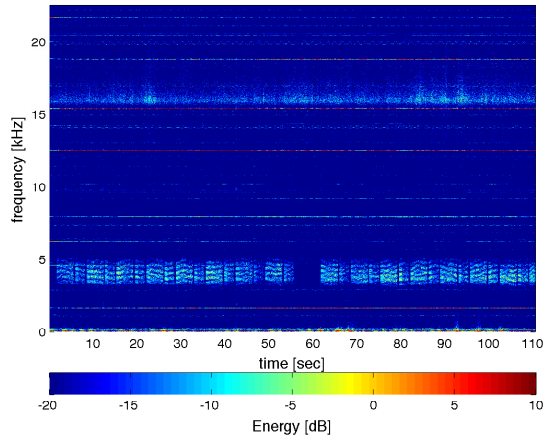
You can see from the spectrogram that there is some significant lower frequency noise coming in on the pressure sensor that is not present on the velocity sensors. There is also some high frequency noise centered around 15 kilohertz. It is also interesting to note that when comparing the x- and y- direction accelerometers that the peaks in energy between each of the directions seem to alternate. In other words where the intensity is highest in the x-direction it is lowest in the y-direction. To verify this observations we integrated the energy in the signals between 2.5 and 5.5 kilohertz to ensure that we are only getting the energy from the signal and not the noise. Figure 5-4 shows the results of this which verified that there are segments of the signals in the x- and y-directions where most of the energy from the signal



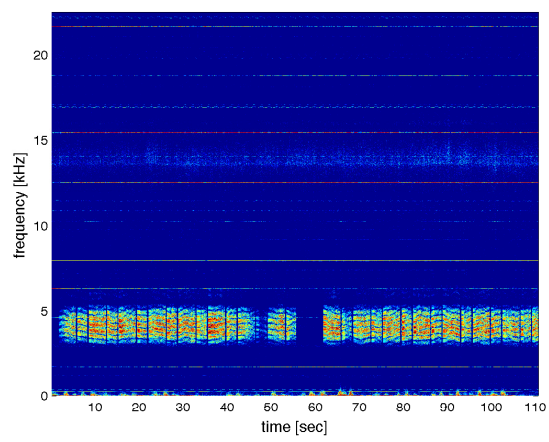
(a)



(b)



(c)



(d)

Figure 5-1: Spectrograms from segment 86: (a) Pressure, (b) X-velocity, (c) Z-velocity and (d) Y-velocity

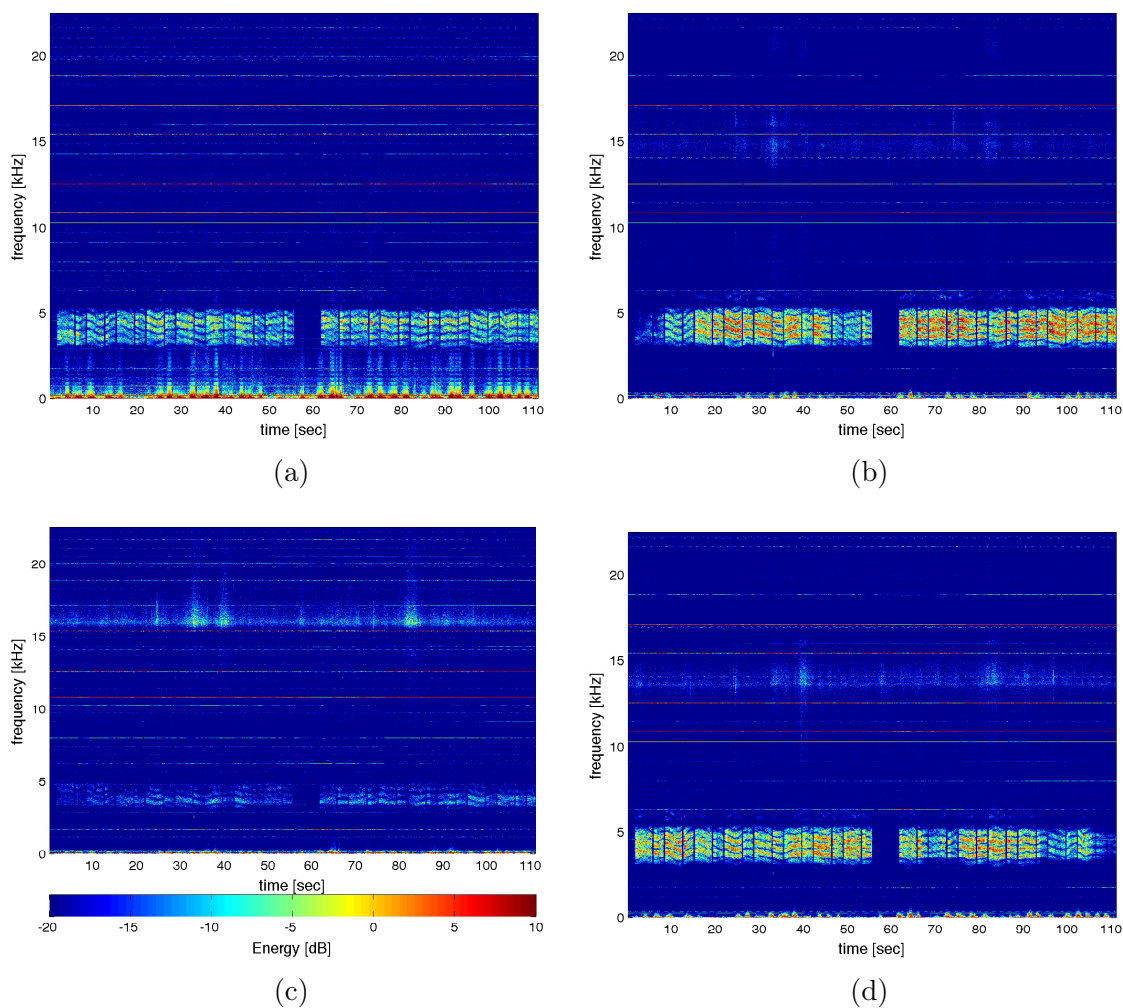
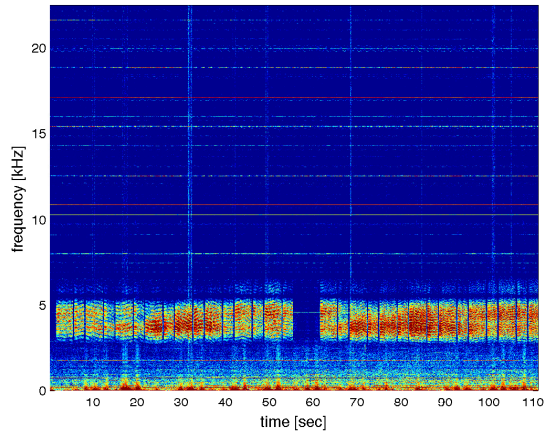
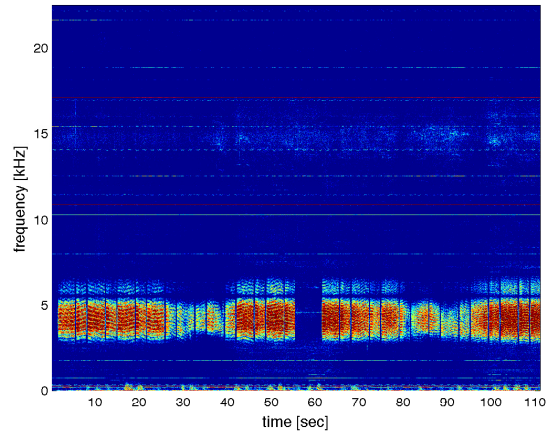


Figure 5-2: Spectrograms from segment 101: (a) Pressure, (b) X-velocity, (c) Z-velocity and (d) Y-velocity

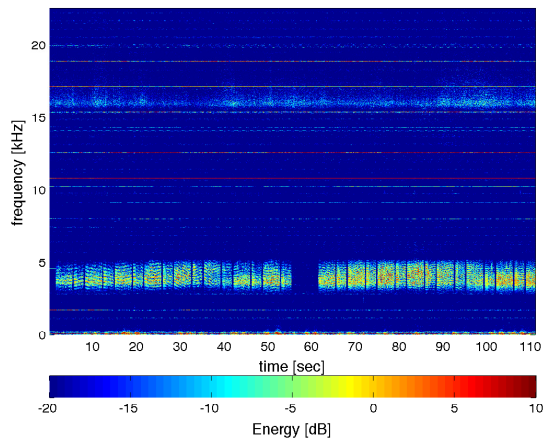




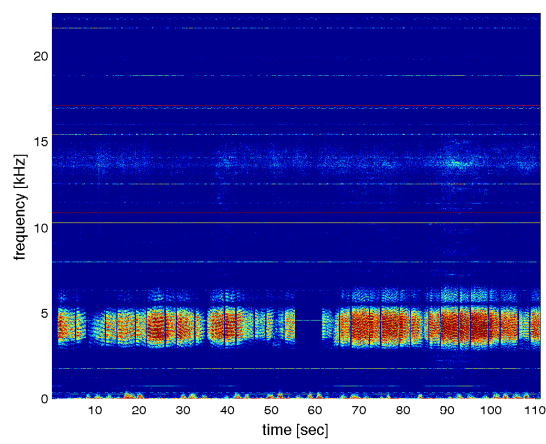
(a)



(b)



(c)



(d)

Figure 5-3: Spectrograms from segment 104: (a) Pressure, (b) X-velocity, (c) Z-velocity and (d) Y-velocity

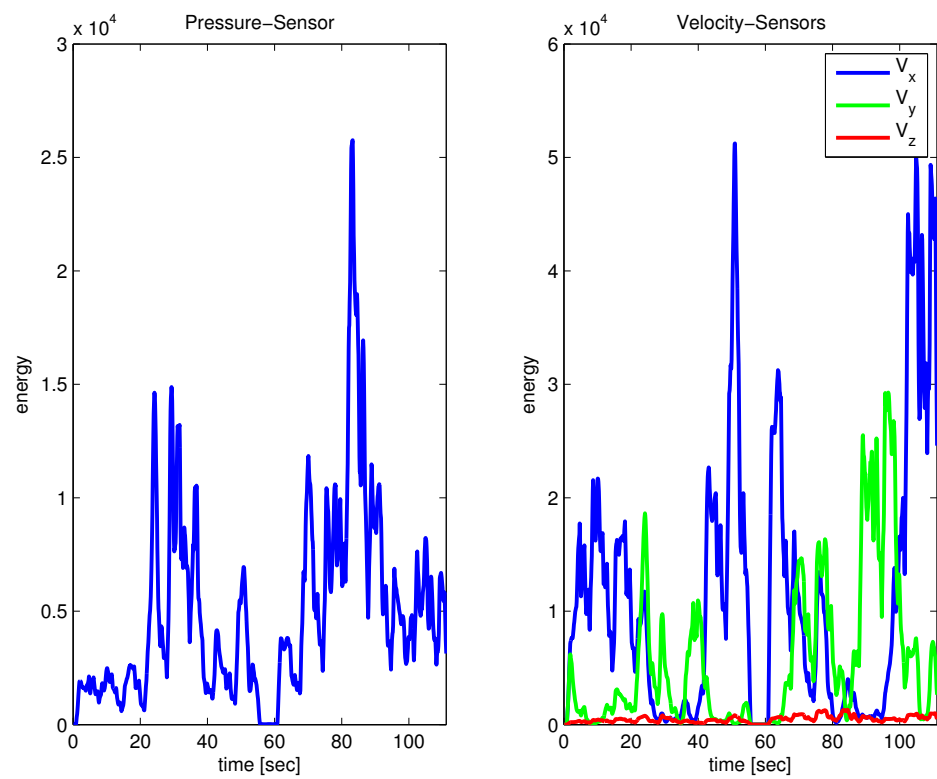


Figure 5-4: Energy in the band of 2.5-5.5 kHz from segment 104 of received data

Segment	Pressure	X-Velocity	Y-Velocity	Z-Velocity
86	29.65	17.75	25.42	14.86
101	22.45	21.66	21.69	13.47
104	33.60	29.52	28.77	25.54

Table 5.1: Average signal-to-noise ratio of data segments expressed in dB

is contained in the x-direction while little is in the y-direction and vice versa. This gives credence to the idea that the beamformed velocity will be more effective in equalizing that the segments separately.

All of the colorbar ranges of each of the spectrogram figures are set to the same range so these figures also give us insight into the signal levels in each of the segments. These figures inform us that the highest native SNR will be seen in segment 104 and that the SNR in segment 86 and segment 101 are roughly similar. The insight that these figures gave was confirmed where the average SNR for each segment can be seen in table 5.1.

Due to unknown technical reasons both sensor 2 and sensor 3 from the VSA did not properly record the received data. Thus we will only be utilizing sensors 1 and 4 for further analysis and the inter-element spacing of the array became 1.00 m ( $\lambda/2 \approx 750$  Hz). Due to the malfunction of the two sensors we were no longer able to perform any multiple sensor beamforming due to aliasing concerns.

The received signals were unpacked and then the DC noise was removed from the signal. Subsequently the signals were then resampled from the sampling rate of the VSA at 45045 hertz to 100 kilohertz and base-banded from the carrier frequency of 3995 hertz. At this point the signal was filtered using the baseband pulse to remove any replicas from the frequency spectrum and then downsampled again to a fractional sample rate of 2 samples per symbol. Finally these signals were matched filtered using the detect data sequence described in section 4.4.

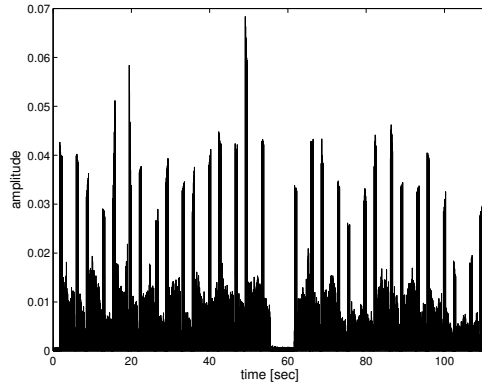
Using the matched filter output we were able to isolate the start of each segment of the received signal as shown in figure 5-5. You will notice in this figure that you can clearly see the start of each segment. You can also see in this figure that there are only 15 full segments, this is due to the fact that the transmitted segments were two minutes long and each of the recorded segments were only 110 seconds. Additionally, using this matched filter output we were able to synchronize each segment to the start of the transmission for subsequent use in the equalizer. From each data file this gives us a total of 15 full signal data segments of Nyquist rate data that can be utilized.

### 5.3 Method

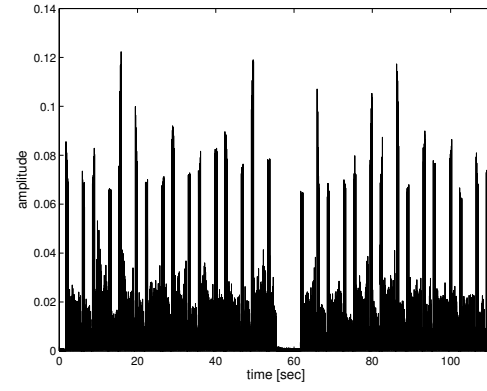
Our method utilizes the directionality of the vector sensors to enable us to determine the optimum steering angle on the horizontal plane and then beamform the x and y velocities to that direction to reduce the noise present in the signal. The first step in this process is to determine the optimum angle at each time for the specific signal. This beamforming was accomplished by taking the base-banded and filtered signal, specifically the x and y velocity portions and applying the following equation to determine the corrected velocity signal where  $v_k$  is the specified velocity signal (x, y or corrected) and  $\phi$  is the horizontal angle.

$$v_{corr} = v_x \cos \phi + v_y \sin \phi \quad (5.1)$$

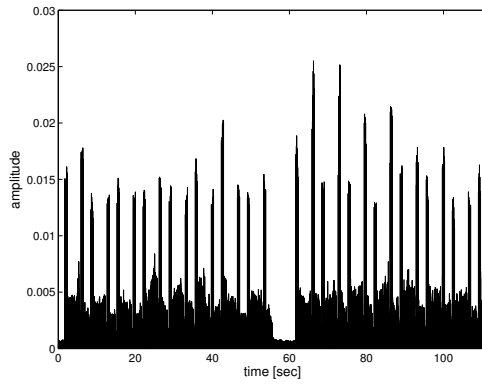
This was performed over a range of angles from 0 to 360° and then integrated the energy present at an angle over a period of one second to determine the optimum beamformed angle for each second. More clearly the optimum beamformed angle was the one where the most energy was present in that time period. Earlier we spoke



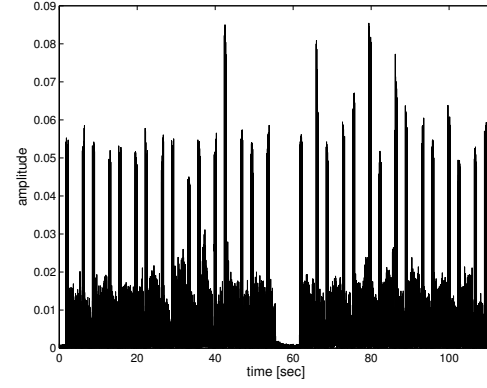
(a)



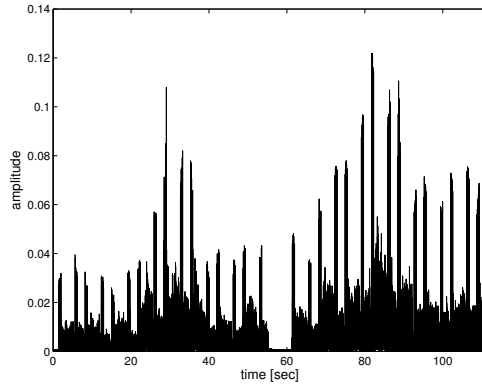
(b)



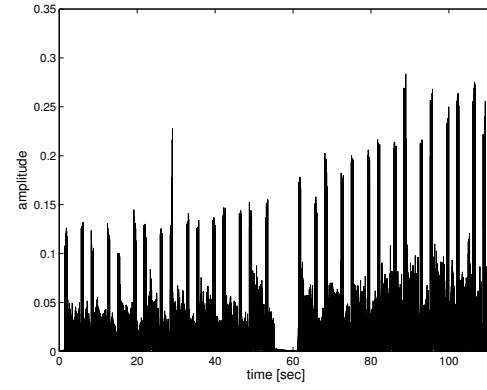
(c)



(d)



(e)



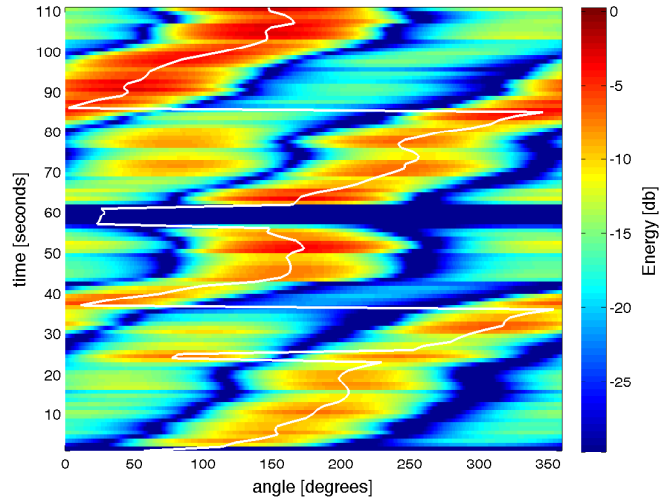
(f)

Figure 5-5: Matched filter output of base banded, filtered and down sampled received pressure signal of (a) segment 86 - sensor 1, (b) segment 86 - sensor 4, (c) segment 101 - sensor 1, (d) segment 101 - sensor 4, (e) segment 104 - sensor 1 and (f) segment 104 - sensor 4

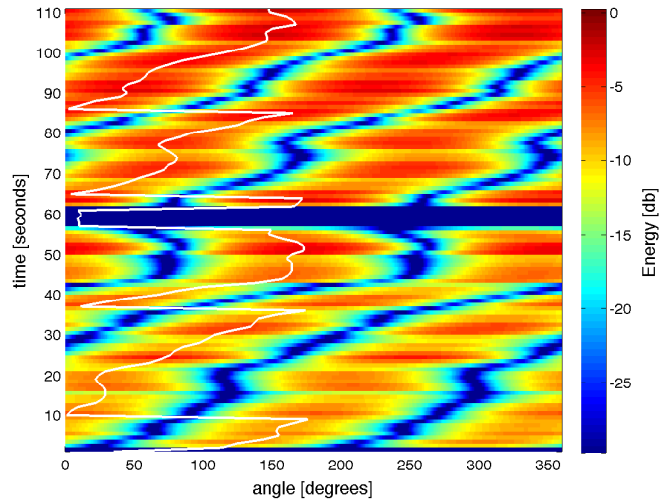
about the  $180^\circ$  ambiguity of the velocity sensors the previous step is the one in which we utilize the pressure sensor to resolve that ambiguity. We performed this step by taking the output of equation (5.1) and comparing the sign of  $v_{corr}$  to the sign of the pressure signal at the same point as discussed in [28] . This tells us whether the acoustic signal was in compression or expansion during that particular sample. If the sign was the same as the pressure signal then the  $v_{corr}$  was left as is, if the sign was opposite  $v_{corr}$  was set to zero to remove the ambiguity and ensure that it would not contribute to the integration of the energy. Figure 5-6 shows a side by side plot of the resolution of the velocity sensor when you correct to resolve the ambiguity and when you do not. You can clearly see from this plot that if you do not correct for the ambiguity it would be easy to steer to the ambiguous angle where you would be steering away from the signal rather than towards it.

By then using a simple routine to determine the angle with the maximum energy at each time interval we now know the optimum steering angle. This optimum angle can be seen in figure 5-7 where you can see that our methodology is able to resolve the ambiguity correctly nearly all the time. However, you can see the one period of time where the optimum angle was improperly determined, this also occurred over one period in figure 5-6a. We did not manually correct for these errors but allowed them to carry through our analysis to show the effect of determining the incorrect angle which is shown in section 5.5

The expansion of this methodology into three dimensions would be very similar except that instead of just a range of angles in the horizontal plane you would also have to include the range of angles from  $0$  to  $180^\circ$  in the vertical plane. Additionally the formula for the corrected velocity would change with the addition of the vertical component to what is represented in equation (5.2), where  $\phi$  is the horizontal angle



(a)



(b)

Figure 5-6: Array resolution of sensor 4 from segment 104 (a) shows the array resolution of the velocity sensor in the horizontal plane when resolving the ambiguity while (b) does not resolve the ambiguity

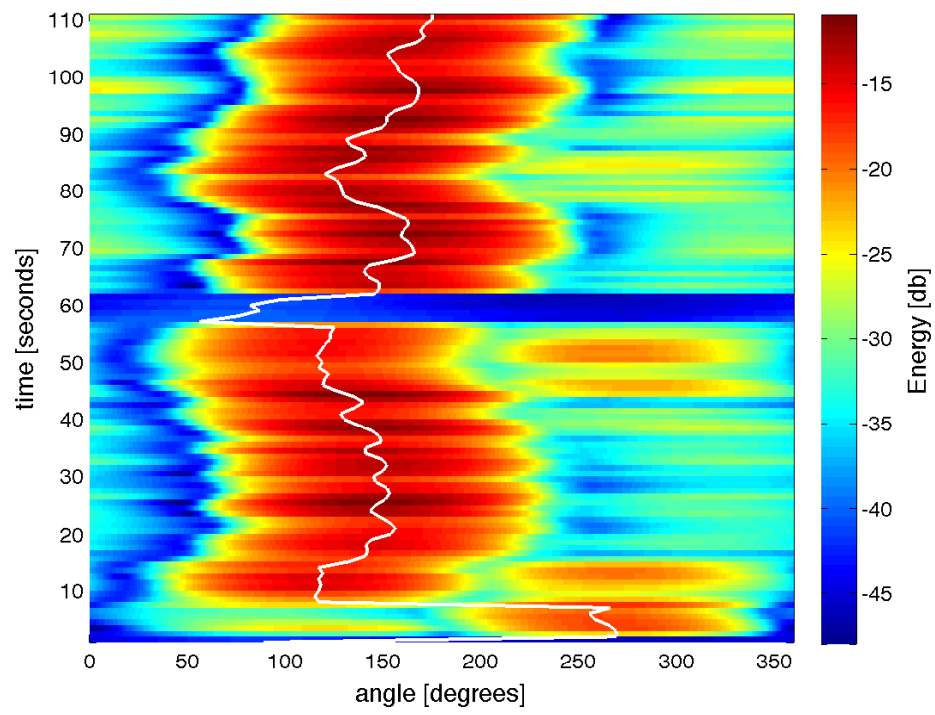


Figure 5-7: Array resolution of segment 101, sensor 1 in the horizontal plane of the velocity sensor with the chosen optimum angle overlaid in white



and  $\theta$  is the vertical angle.

$$v_{corr} = v_x \sin \theta \cos \phi + v_y \sin \theta \sin \phi + v_z \cos \theta \quad (5.2)$$

Resolving the ambiguity would occur in the exact same manner utilizing the pressure sensor to determine which direction the signal is actually arriving from. We expanded our optimum angle determination to three dimensions and the results from this for an individual sensor is shown in figure 5-8. This figure shows that for this segment the

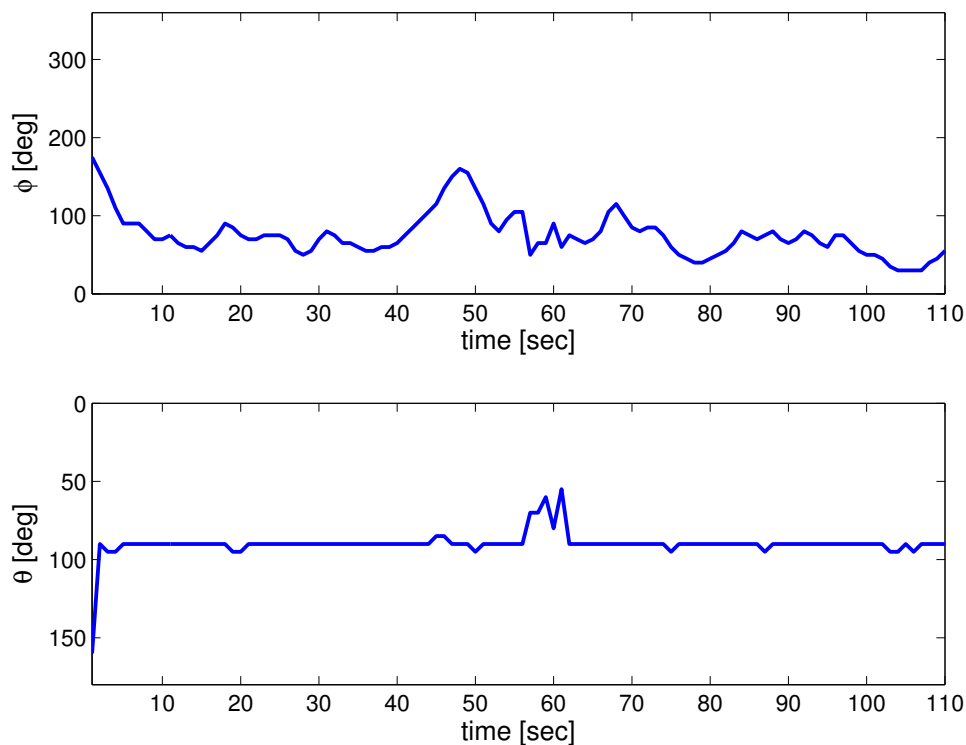


Figure 5-8: Optimum angles for sensor 1 of segment 86. The upper plot shows the optimum horizontal angle ( $\phi$ ) and the bottom plot shows the optimum elevation angle ( $\theta$ )

optimum elevation angle is the horizontal plane that our in-depth analysis occurs in, this was consistent through all of the segments. This also gives us the insight that our

algorithm is finding the maximum energy because during the period where the angle determination was meaningless and there is no signal being received the optimum elevation moves towards the surface where the loudest noise would be coming from. The scope of this thesis does not include an in-depth analysis of three-dimensions and is a topic for further study.

## 5.4 Equalization Parameters

In analyzing our results when considering the equalization of the signal it is important to remember that we are operating in the training mode of the equalizer. This means that at each step it is comparing the output of the equalizer to the actual transmitted signal at that point and then feeding back that correct signal into the feedback portion of the equalizer. We analyzed a number of different combinations of filter length, RLS forgetting factor ( $\lambda$ ) and NLMS step-size ( $\mu$ ) to determine the optimum values for the processing of the data. When processing the native and degraded SNR signals we also used different combinations of the data in a multi-channel equalizer to analyze the benefits of using beamforming even further.

### 5.4.1 Filter Length

In determining the filter lengths we used a simple method of trial and error to optimize the results. The only requirement that we imposed was that the filter was long enough to include the entire impulse response of the channel which was found to vary from  $\approx 2.0$  msec to 3.5 msec which corresponds to a filter length of 6 - 9 symbols where our feedforward filter length was 15 symbols or 30 samples. Our feedback filter was chosen to be 20 symbols. Optimizing not only the feedforward

but also the feedback filter lengths is an area that is in need of further study.

### 5.4.2 Forgetting Factor and Step Size

The forgetting factor ( $\lambda$ ) determines an effective length of the window of time that the RLS algorithm averages the observations to calculate the equalizer weights. When referring to the averaging window length it is approximated by  $\frac{1}{1-\lambda}$ . It is best to set this much less than the coherence length of the channel. This was initially analyzed using a specific NLMS step size to get an idea of the range of "good" averaging window lengths. An example of this can be seen in figure 5-9, this figure utilized a NLMS step size of 0.0012.

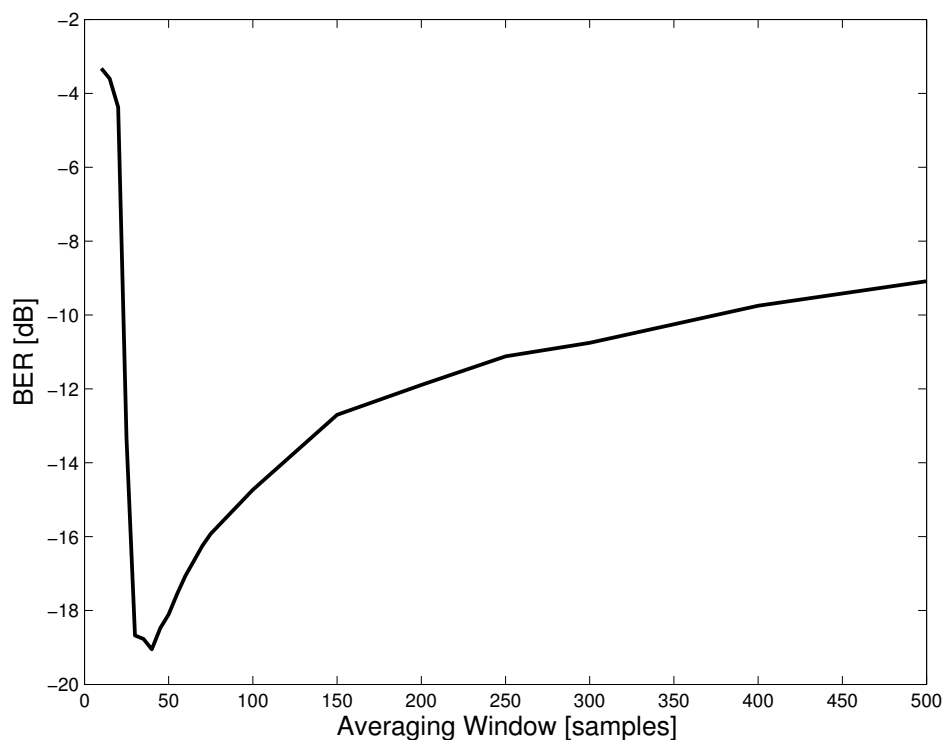


Figure 5-9: Equalizer BER performance for different averaging window lengths

From figure 5-9 you can see that optimum range of averaging windows is approximately 30 - 60 samples which correspond to forgetting factors ranging from 0.9667 to 0.9833. You can also see that there is a narrow window where the optimum lengths lie and if you go lower the error rate will increase significantly. The reason for this is that the RLS adaptation algorithm needs enough samples in the averaging window to make reasonable estimates of the correlation matrix and vector required to estimate the optimal filter length. Conversely, if you incorporate a longer averaging window the error rate will go up but at a much slower rate. The reasoning behind this is as you increase the averaging window length beyond the coherence time of the channel the equalizer can not accurately track the fluctuations in the channel. The determination of the forgetting factor also gives us insight into the behavior of our channel, the shorter the optimum averaging window the quicker the variations in the channel are occurring.

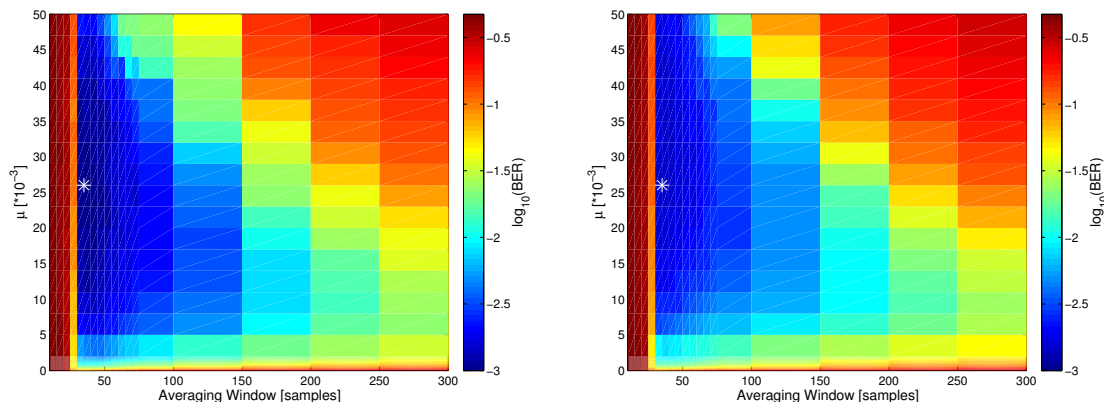
With these values in mind we performed an expansive analysis of all fifteen data segments to determine the appropriate forgetting factor and step sizes for use in our further analysis. From the initial trial we had a good idea of what the appropriate averaging window would turn out to be; for thoroughness we expanded the window to evaluate for a range of  $\lambda$  values from 0.9000 to 0.9980, this corresponds to averaging window lengths of 10 to 500 samples.

To determine the NLMS step size we analyzed for a range of values from 0 to 0.05. Using this range of values we ran the single-channel equalizer on the pressure signal, a multi-channel equalizer on the pressure signal combined with the x and y velocity signal and finally a multi-channel equalizer on the x and y velocity signals only. These runs were performed on the signals received on both sensor 1 and 4 to determine the optimum step size and forgetting factor. The z-velocity signal was not utilized in these runs because its inclusion in a multi-channel equalizer

caused the equalizer performance to degrade by more than an order of magnitude. Representative examples of the results from this analysis are shown in figure 5-10. Table 5.2 shows what the optimum was determined to be for each of the different signal combinations.

Sensor	Input	Optimum Averaging Window Length ( $\lambda$ )	Optimum Step-size ( $\mu$ )
1	$P$	20 ( $\lambda = 0.9500$ )	$\mu = 0.014$
	$V_x, V_y$	25 ( $\lambda = 0.9600$ )	$\mu = 0.014$
	$P, V_x, V_y$	35 ( $\lambda = 0.9714$ )	$\mu = 0.026$
4	$P$	20 ( $\lambda = 0.9500$ )	$\mu = 0.023$
	$V_x, V_y$	25 ( $\lambda = 0.9600$ )	$\mu = 0.011$
	$P, V_x, V_y$	30 ( $\lambda = 0.9667$ )	$\mu = 0.026$

Table 5.2: Optimum  $\mu$  and  $\lambda$  values for data combinations analyzed



(a) Segment 104 - Sensor 1:  $P, V_x$  and  $V_y$       (b) Segment 104 - Sensor 4:  $P, V_x$  and  $V_y$

Figure 5-10: Examples of interpolated results to determining optimum  $\lambda$  and  $\mu$  values. The optimum in each run is shown by the white star and explicitly stated in table 5.2

Using this information we chose a forgetting factor of  $\lambda = 0.9667$  which corresponds to an averaging window length of 30 and a NLMS step-size of  $\mu = 0.0260$ .

The larger averaging window was chosen due to the fact that; as we discussed above; if your window is too short you will not have enough information to properly model the channel. By choosing a global averaging window to be slightly longer while we may have very slightly degraded our error rate in the pressure only and velocity only runs we will have better overall results. Additionally, the window lengths are so close to each other that computational expense does not factor in here. Finally, the same averaging window length was used for all of the SNRs.

## 5.5 Results

As a review the overall methodology that we utilized is encapsulated in figure 5-11. In this figure the basebanded, filtered and down-sampled received data is the raw data. This data then has its SNR adjusted to the appropriate level while in parallel the array resolution is computed and the optimum angle is determined. This optimum angle is then used to beamform the velocity and then the combined velocity and/or pressure are entered into the TU-RLS algorithm. The output of the TU-RLS algorithm is then fed into the decision device where the decisions is compared to the transmitted data to compute the bit error rate. Since our TU-RLS algorithm was only operated in training mode the transmitted data is also an input into the equalizer as the feedback data and as the desired signal for the adaptation process.

### 5.5.1 Signal-to-Noise Ratio

The first place to analyze our results is how the SNR of the signal is affected by beamforming. Figure 5-12 shows the SNR levels for all of the segments that were analyzed for this thesis. You can see from this figure that the beamformed velocity

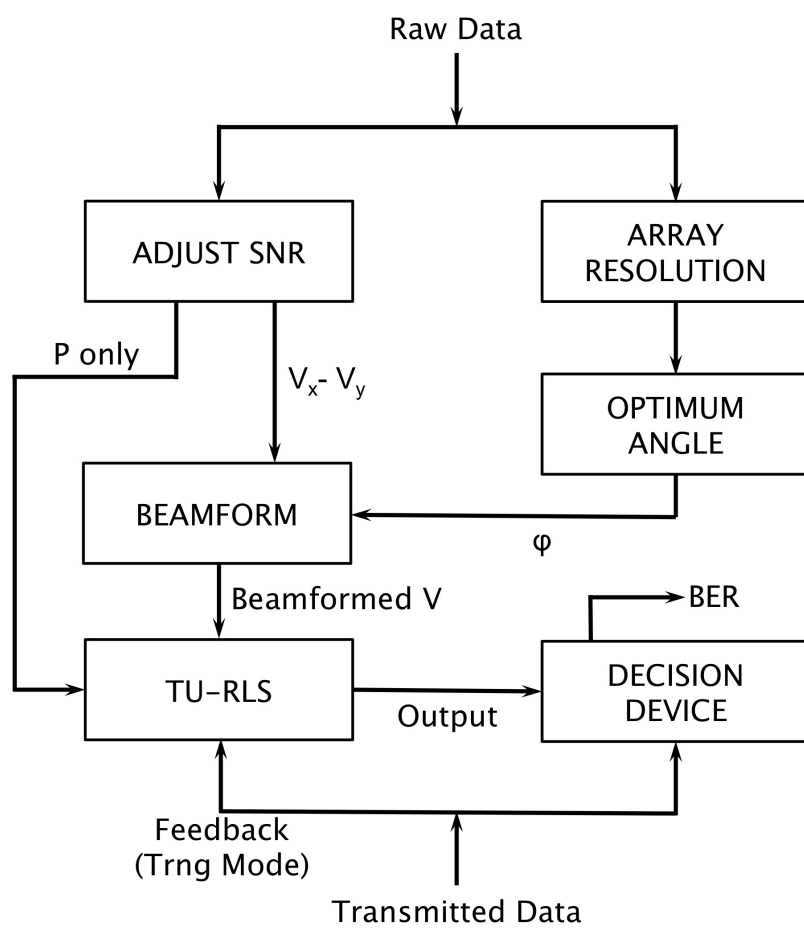
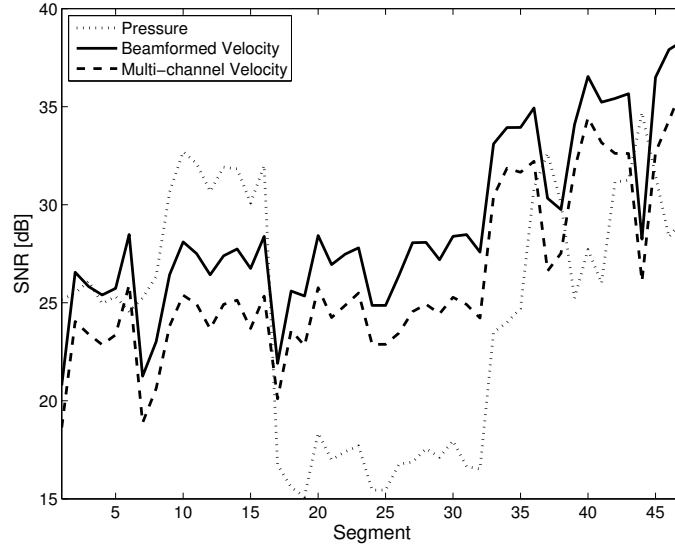
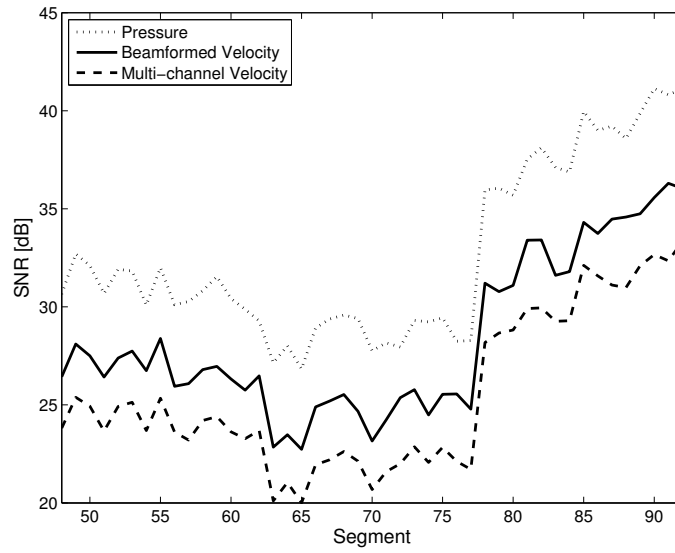


Figure 5-11: Block diagram of our processing methodology



(a)



(b)

Figure 5-12: SNR of velocity components of each segment analyzed for this thesis (a) shows the segments received on sensor 1 and (b) shows the segments received on sensor 4



showed an average improvement of 2.7 dB, with a maximum improvement of 3.9 dB and a minimum improvement of 1.8 dB over the SNR of the multi-channel velocity. Figure 5-13 shows a more detailed segment of data where you can see that the SNR of beamformed velocity is an improvement over not only the multi-channel but also each of the velocity components. This tells us by steering the velocity components

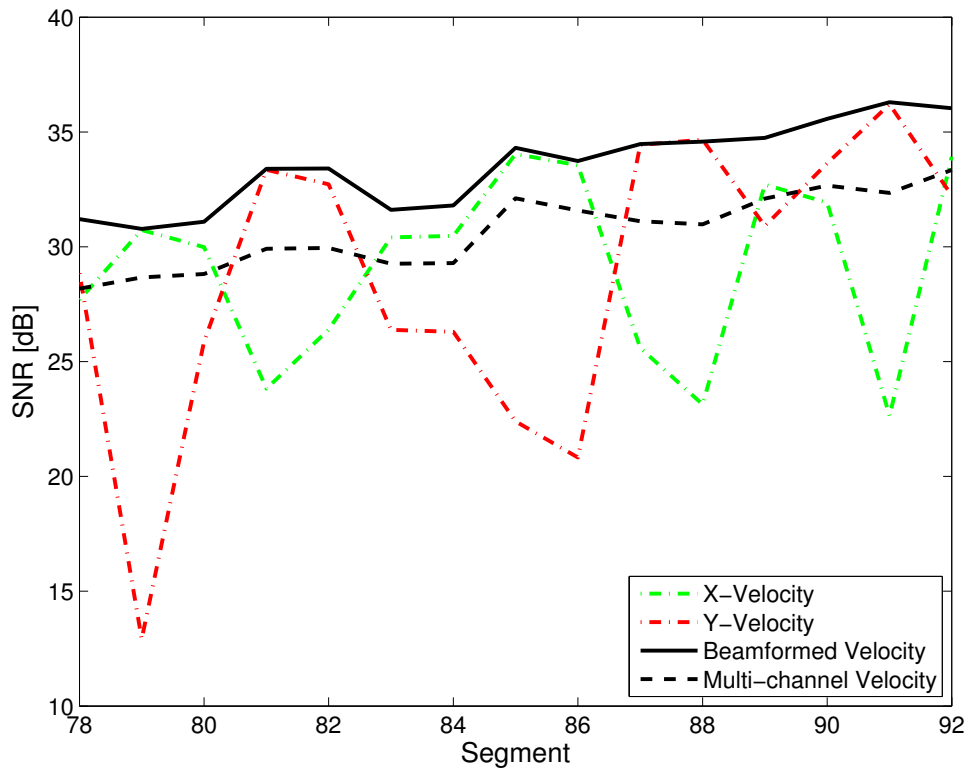


Figure 5-13: Comparison of the SNR of beamformed velocity with multi-channel velocity and its components

to the proper angle we are removing the effect of a significant portion of the noise that will subsequently result in improvements in the equalization of the signal.

### 5.5.2 Adaptive Equalization

When exploring the gain that beamforming the velocity signals provides we explored four different equalizations at the native and degraded SNRs. We used as our baseline the adaptive equalization of only the received pressure signal in a single-channel equalizer. All other results have been normalized to this baseline to show the improvement achieved. The other three equalizations that we explored were:

- X- and Y-velocity signals in a multi-channel equalizer
- Beamformed velocity signal in a single-channel equalizer
- Beamformed velocity signal and pressure signal in a multi-channel equalizer

The results of these runs are shown in figures 5-14, 5-15, 5-16 and 5-17 which represent the native SNR, 20 dB, 10 dB and 0 dB respectively. The results are plotted as a cumulative distribution function (CDF) where the vertical axis represents the percentage of the number of segments that performed better than the ratio on the horizontal axis. The horizontal axis represents the ratio of the bit error rate,  $BER_{test}$ , of the method under consideration to the bit error rate,  $BER_p$ , achieved by equalizing just the pressure signal as shown in equation (5.3).

$$\frac{BER_{test}}{BER_p} \tag{5.3}$$

When comparing each of the three equalizers at the different noise levels we examined how they performed with respect to each other as well as compared to the other noise levels. In each of the four noise levels the beamformed velocity had better overall performance. This is represented by the beamformed CDF curve being to the left of the  $V_x$ - $V_y$  CDF meaning that at each of the ratios a high percentage of

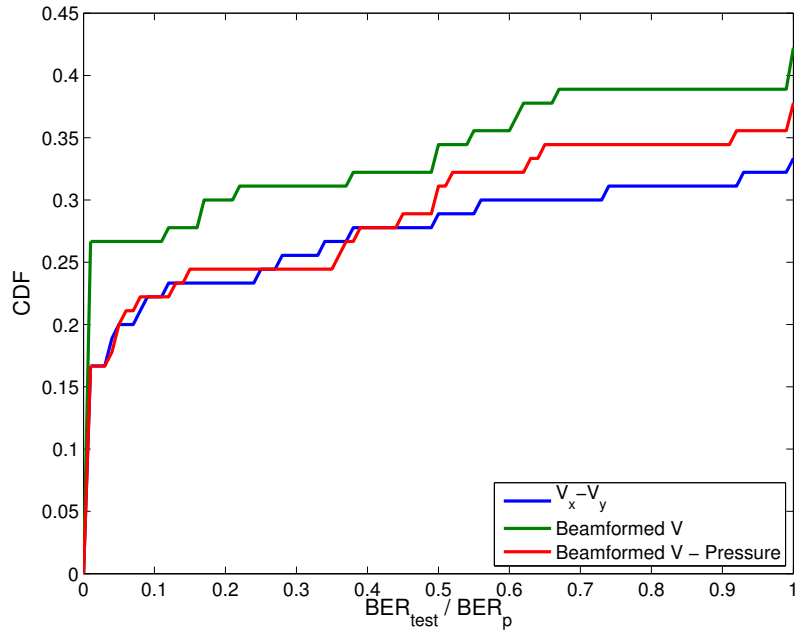


Figure 5-14: CDF of BER for native SNR

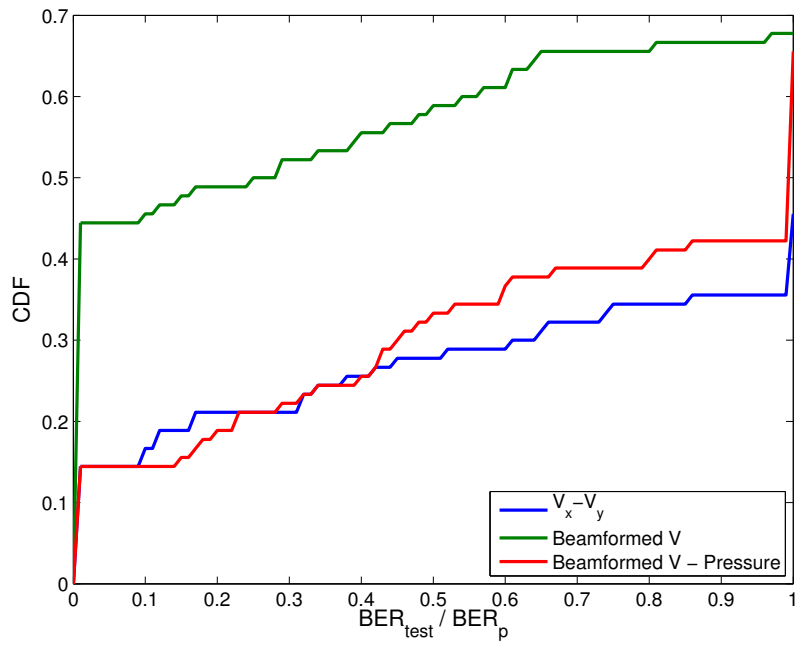


Figure 5-15: CDF of BER for SNR = 20 dB

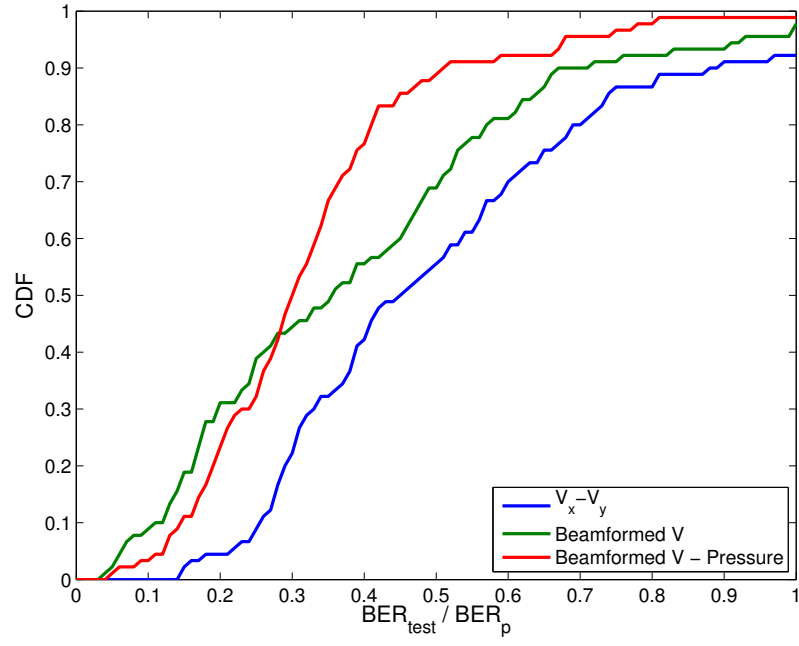


Figure 5-16: CDF of BER for SNR = 10 dB

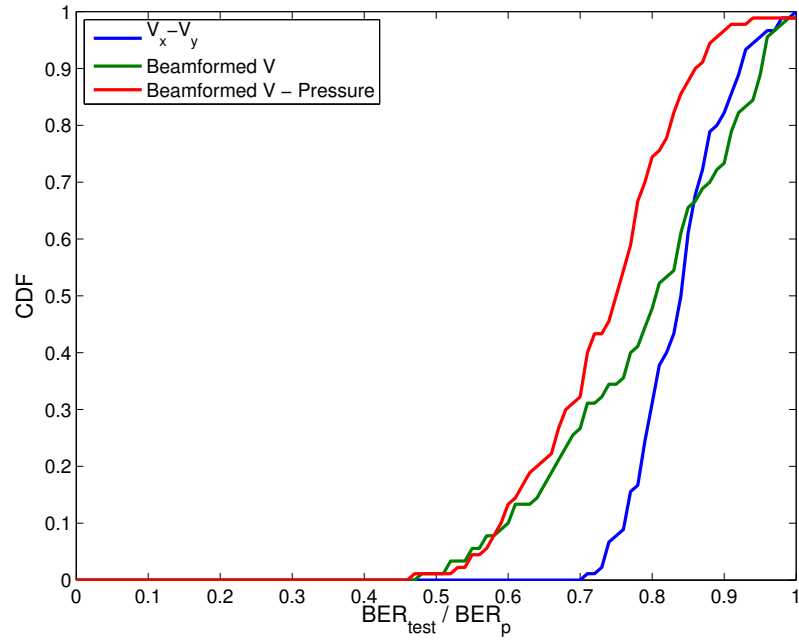


Figure 5-17: CDF of BER for SNR = 0 dB

the segments performed at that level in the beamformed equalizer versus the  $V_x$ - $V_y$  multi-channel equalizer. At the higher SNR levels the beamformed velocity not only outperformed the  $V_x$ - $V_y$  but also the multi-channel equalizer with the beamformed velocity and the pressure. However, as the SNR degraded the beamformed velocity and pressure multi-channel equalizer performed at an equivalent level or better than the beamformed velocity on its own.

Interestingly, at the higher signal-to-noise ratios (native and 20 dB) the equalizer using just the pressure signal performed significantly better than both the beamformed and multichannel equalized signals. At native SNR, the pressure sensor outperformed all of the vector sensor and pressure combinations in more than half the situations. Similarly, at a SNR of 20 dB it outperformed the beamformed velocity in at least 30% of the situations and in at least 50% of the situations outperformed the multi-channel  $V_x$ - $V_y$  and multi-channel beamformed velocity and pressure. It can also be seen that as the SNR degrades nearly 100% of the time the vector sensor combinations all outperform the pressure only equalizer.. For example when examining the beamformed velocity-pressure multi-channel equalizer at native SNR approximately 22% of the segments outperform the pressure sensor tenfold while at an SNR of 20 db only 15% outperform the pressure sensor by the same amount. When degrading the SNR even further to 10 dB approximately 4% of the segments perform at that level and at an SNR of 0 dB approximately 1% perform 2 times better than the pressure only sensor.

### 5.5.3 Computational Cost

Another important result that can be seen from these plots is that equivalent or better results can be achieved from a more efficient equalizer. The primary computational

cost from equalization comes from the inversion of the sample covariance matrix, in the LMS case this is an  $O(N^2)$  operation and once it is implemented through the RLS family of algorithms it becomes an  $O(N \log N)$  operation. The size of the sample covariance matrix is determined by the length of the feedforward and feedback filters. As discussed in section 2.3.5 when you implement a multiple channel you are simply stacking the input vectors for each channel in the feedforward portion of the equalizer. If you consider a single channel decision feedback equalizer (DFE) with feedforward filter length of  $N$  and feedback length of  $M$  the size of the covariance matrix will be  $(N + M) \times (N + M)$ . Now considering a  $n$ -channel decision feedback equalizer with the same feedforward and feedback filter lengths the size of the covariance matrix would be  $(nN + M) \times (nN + M)$ .

With that knowledge and considering the results of our equalization we can see that effect that beamforming the velocity has besides just the improvement in the performance of the equalizer. When comparing the performance of the beamformed velocity DFE with the multi-channel velocity DFE you are not only seeing an improvement in the performance of the equalizer but also a savings in the computational cost of the operation. Taking it one step further we saw that at lower signal-to-noise ratios the multi-channel DFE with beamformed velocity and pressure completely outperforms the multi-channel velocity DFE and performs either equivalently or better than the single channel beamformed velocity DFE. From this we can draw that for the same computational cost as the multi-channel velocity DFE you can have a multi-channel DFE that incorporates both the velocity elements and the pressure element.

This savings becomes even more significant when this application is extended into three dimensions. Instead of a three-channel velocity DFE you can utilize a single channel DFE that incorporates all of the same data; all three velocity signals; or

a two-channel DFE that incorporates all of the data the vector sensor can provide; pressure sensor and all three velocity sensors.





# Chapter 6

## Conclusions

### 6.1 Summary and Conclusions

As we have discussed adaptive equalization is an important component in the field of underwater acoustic communication systems. Just as important as the adaptive equalization is the sensor component of the field. In this thesis we considered whether or not vector sensors can be better utilized in underwater acoustic communications through beamforming the elements of the vector sensor. We have presented an array of different results and insights into this question which we will briefly summarize.

We initially went over the basic physics and theoretical concepts that drive the use and operation of vector sensors. We discussed the ambiguity that vector sensors suffer from as well as the incorporation of the pressure data to resolve that ambiguity in real time which is an improvement over traditional pressure-sensor ambiguity techniques. In a discussion about adaptive equalization we discussed the use of the method of least squares and why the optimal detectors (such as a matched filter) were not optimal in our case. We then discussed the implementation of the method

of least squares (LMS) through the computationally efficient recursive least squares algorithm (RLS). Finally, this was expanded to account for doppler spreading in the channel utilizing the time-update recursive least squares algorithm (TU-RLS).

In order to better understand the environment that the KAM11 vector sensor array (VSA) experiment was conducted in we performed some modeling using BELLHOP, a ray trace method modeling code. We briefly went over the basics of how a ray trace model works and some of the specifics that are implemented through BELLHOP and the effect they had. We had to make very broad assumptions in this modeling about the sound speed profile in the region in which the experiment was conducted due to equipment issues. However we were still able to make some valid conclusions that were later verified by the received data.

- Based on the geometry of the experiment we expected two arrivals at the VSA; one from direct path and another from a surface bounce interaction. From this we were also able to approximate the delay in the arrival times.
- We were able to see that that arrivals were expected to arrive at the array at shallow angles (near horizontal) which allowed us to make the assumption that optimizing over the horizontal plane would give us accurate results.
- From the transmission loss runs we expected that both arrivals would have similar energy levels and one peak would not significantly dominate the other.

We then discussed the KAM11 experiment and the physical geometry of the vector sensor array as well as the set-up of the trial. We then delved into a description of the time variability of the channel that forces us to utilize adaptive equalizers rather than optimum filters. After discussing the construction of the transmitted signal we also discussed the additive noise segment that was used to degrade the received

signal as well as the method that was used to add this noise.

After analyzing the received data and the noise characteristics in the files we then proceed to beamform the velocity components of the signal to determine the optimum angles. Utilizing this data set we were able to perform adaptive equalization of the received signals in four configurations: single-channel pressure, multi-channel velocity, single-channel beamformed velocity and multi-channel with pressure and beamformed velocity. From this analysis we were able to make the following conclusions:

- The SNR of the beamformed velocity is on average approximately 3 dB greater than that of the multi-channel velocity. This is a significant gain and is due to beamforming steering the "view" of the vector sensor to the direction of highest energy and thus minimizing the effect of the noise coming from all of the other directions.
- Steering the beam on a slow time scale allows us to exploit all of the incoming data in a more computationally efficient manner to achieve better performance in an adaptive equalizer. While it was not shown in this thesis our results indicate that further study will show that in three dimensions you are able to achieve even better performance with additional savings in computational expense.
- The ambiguity of the vector sensor can be properly resolved utilizing the pressure sensor element of the vector sensor. The performance in this regard was not perfect and was actually found to perform slightly better at lower native SNR. This was only our experience in this analysis and without further study that conclusion could not be applied globally. Additionally, it is important to note

that in the segments where the ambiguity was not properly resolved the performance of the equalizer suffered significantly.

- If it can be shown that the arrivals of the signals are at smaller angles of elevation that in most cases that optimum elevation angle will be horizontal. This insight can be utilized to simplify the optimization of the angles. This was found to be the case when expanding the optimum angle selection to three-dimensions but further study would be required to verify this across more than our experiment.

Overall, the work in this thesis showed the vector sensors can absolutely be utilized in underwater acoustic communications via adaptive equalization. Additionally with proper steering on a slow time-scale the vector sensor in most cases will provide performance improvement while saving on computational cost.

## 6.2 Future Work

We have shown that a properly steered vector sensor will display a performance improvement in adaptive equalization but no rigorous mathematical explanation was given as to why or how this occurs - we merely utilized our intuition as justification. Further study is required to determine why this improvement is seen through use of our method.

Additionally we focused only on steering in the horizontal plane with some short forays into three-dimensions to determine the optimal elevation. This methodology must be expanded fully into three dimensions to verify our intuition that we will see continued performance improvement along with additional computational savings. Along these same lines further experiments need to be conducted over longer ranges

so that there will be bottom bounce arrivals so that the optimal elevation angles are not only near the horizontal plane and it will be necessary to utilize the z-axis velocity sensor information. Part of this continued analysis in three dimensions will require further study of the computational savings and what the optimum configuration is to achieve the best performance with the lowest computational expense.

We utilized omni-directional noise as our additive noise and saw no real directional noise sources that we could distinguish in the signals. Therefore our optimum angle method never had to discriminate between a lot of noise in one direction and the signal in another. This is an area that would be useful to explore further and add robustness to our methodology. In the case where the noise and the signal are in the same direction our expectation would be that the optimum angle would still be chosen properly but we would see a degradation in the performance of the equalizer, this also needs to be explored further.

In terms of the entire process an avenue that absolutely must be examined is the expansion of these methods to real-time, this would also necessitate taking the DFE out of training mode and determining the best methodology for making our decisions. Concurrently with that expansion we also need to further our exploration into array-based processing using these techniques. While we had a string a vector sensor elements deployed we could not perform this analysis due to the failure of two elements which raised aliasing concerns.

Finally, a minor area that would benefit from further study is a method to determine the optimum equalization parameters of (1) feed-forward filter length, (2) feedback filter length, (3) LMS step-size ( $\mu$ ) and (4) averaging window length ( $\lambda$ ). As our method for choosing the LMS step-size and averaging window length utilized brute force computation and the filter lengths we merely tweaked using our intuition to obtain the best results.



# Bibliography

- [1] D. H. Brandwood. A complex gradient operator and its application in adaptive array theory. *Microwaves, Optics and Antennas, IEE Proceedings H*, 130(1):11–16, February 1983.
- [2] J. M. Cioffi, W. L. Abbott, H. K. Thapar, C. M. Melas, and K. D. Fisher. Adaptive equalization in magnetic-disk storage channels. *Communications Magazine, IEEE*, 28(2):14–29, February 1990.
- [3] T. H. Eggen, A. B. Baggeroer, and J. C. Preisig. Communication over Doppler spread channels - part I: channel and receiver presentation. *Oceanic Engineering, IEEE Journal of*, 25(1):62–71, January 2000.
- [4] P. Fan and M. Darnell. *Sequence Design for Communications Applications*. Research Studies Press LTD., 1996.
- [5] M. Hawkes and A. Nehorai. Acoustic vector-sensor beamforming and capon direction estimation. *Signal Processing, IEEE Transactions on*, 46(9):2291–2304, September 1998.
- [6] S. Haykin. *Adaptive Filter Theory*. Pearson Higher Education, Inc., fifth edition, 2014.
- [7] B. Hochwald and A. Nehorai. Identifiability in array processing models with vector-sensor applications. *Signal Processing, IEEE Transactions on*, 44(1):83–95, Jan 1996.
- [8] W. Hodgkiss and J. C. Preisig. Kaua’i Acomms MURI (KAM11) Experiment. In *European Conference on Underwater Acoustics*, 2012.
- [9] W. Hodgkiss, H. Song, G. Deane, J. Preisig, M. Badiey, and A. Song. Kaua’i Acomms MURI 2011 (KAM11) Experiment. Trip Report, July 2011.

- [10] W. Hodgkiss, H. Song, G. Deane, J. Preisig, M. Badiey, and A. Song. Kaua'i Acomms MURI 2011 (KAM11) Experiment. Log Book, July 2011.
- [11] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt. *Computational Ocean Acoustics*. Modern Acoustics and Signal Processing. Springer Science+Business Media, 2nd edition, 2011.
- [12] D. B. Kilfoyle and A. B. Baggeroer. The state of the art in underwater acoustic telemetry. *Oceanic Engineering, IEEE Journal of*, 25(1):4–27, January 2000.
- [13] J. P. Kitchens. Acoustic vector-sensor array performance. Master's thesis, Massachusetts Institute of Technology, 2008.
- [14] C. B. Leslie, J. M. Kendall, and J. L. Jones. Hydrophone for measuring particle velocity. *The Journal of the Acoustical Society of America*, 28(4):711–715, July 1956.
- [15] A. Nehorai and E. Paldi. Acoustic vector-sensor array processing. *Signal Processing, IEEE Transactions on*, 42(9):2481–2491, September 1994.
- [16] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing. Pearson Higher Education, Inc., 3 edition, 2010.
- [17] A. V. Oppenheim and G. C. Verghese. Signals, Systems, and Inference. Class Notes for 6.011: Introduction to Communication, Control and Signal Processing, May 2013.
- [18] M. B. Porter. The BELLHOP Manual and User's Guide. Preliminary Draft, January 2011.
- [19] J. C. Preisig. Performance analysis of adaptive equalization for coherent acoustic communications in the time-varying ocean environment. *The Journal of the Acoustical Society of America*, 118(1):263–278, July 2005.
- [20] J. C. Preisig. Time Series Analysis notes from Summer 2013. Recorded by M.R. Lewis, June-August 2013.
- [21] J. G. Proakis. Adaptive equalization for TDMA digital mobile radio. *Vehicular Technology, IEEE Transactions on*, 40(2):333–341, May 1991.
- [22] R. Sharma, W. D. Grover, and W. A. Krzymien. Forward-error control (FEC)-assisted adaptive equalization for digital cellular mobile radio. *Vehicular Technology, IEEE Transactions on*, 42(1):94–102, February 1993.



- [23] J. C. Shipps and K. Deng. A miniature vector sensor for line array applications. In *OCEANS 2003*, volume 5, pages 2367–2370. IEEE, 2003.
- [24] M. Stojanovic. Recent advances in high-speed underwater acoustic communications. *Oceanic Engineering, IEEE Journal of*, 21(2):125–136, April 1996.
- [25] M. Stojanovic. Acoustic (underwater) communications. In *Encyclopedia of Telecommunications*. Wiley, 2003.
- [26] M. Stojanovic and J. C. Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *Communications Magazine, IEEE*, 47(1):84–89, January 2009.
- [27] H. L. Van Trees. *Optimum Array Processing. Part IV of Detection, Estimation and Modulation Theory*. John Wiley & Sons, Inc., New York, NY, 2002.
- [28] K. T. Wong and H. Chi. Beam patterns of an underwater acoustic vector hydrophone located away from any reflecting boundary. *Oceanic Engineering, IEEE Journal of*, 27(3):628–637, July 2002.